

1.	Классификация и структура микроконтроллеров.....	3
2.	Микроконтроллер. Основные классификационные признаки МК.	4
3.	Тактовая частота. Производительность. Потребляемая мощность.....	5
4.	Сферы применения микроконтроллеров.....	6
5.	Аппаратная платформа Arduino. Характеристики.....	7
6.	Закон Мура. График.	8
7.	Отличительные признаки МК.....	9
8.	Модульная организация МК. Описание.....	10
9.	Память программ. Память данных.	11
10.	Регистровая память. Энергозависимая память. Оперативная память...	12
11.	Периферия МК. Описание.	13
12.	Цифровые порты ввода/вывода.	14
13.	Входные элементы управления. Схемы подключения.	15
14.	Выходные элементы управления. Схемы подключения.	16
15.	Прерывания. Таймеры/счётчики.....	17
16.	Работа с аналоговыми сигналами. АЦП.	18
17.	Тактовый генератор. Система реального времени.....	19
18.	Последовательный интерфейс UART/USART. Подключение. Формат данных.	20
19.	Последовательный интерфейс SPI. Подключение. Формат данных.....	21
20.	Двухпроводной последовательный интерфейс TWI/I2C. Подключение. Формат данных.....	22
21.	Интерфейс JTAG. Подключение. Формат данных.....	23
22.	Интерфейс CAN. Подключение. Формат данных.....	24
23.	Языки программирования МК. Особенности.	25
24.	Типы корпусов МК. Описание. Назначение.....	26
25.	Анимация. HMI.....	27
26.	Микроконтроллеры STM32. Описание. Характеристики.....	28
27.	Микроконтроллеры AVR. Описание. Характеристики.....	29
28.	Микроконтроллеры ESP8266. Описание. Характеристики.	30
29.	Микроконтроллеры ESP32. Описание. Характеристики.....	31
30.	Сервопривод. Описание. Устройство сервопривода.....	32
31.	Сервопривод. Преимущества. Недостатки. Схема подключения.....	33
32.	Шаговый двигатель. Описание. Принцип работы. Преимущества. Недостатки.	34
33.	Шаговый двигатель. Преимущества. Недостатки. Схема подключения.	35
34.	Гироскоп. Акселерометр. Описание. Назначение.	37
35.	Ультразвуковой датчик расстояния. Принцип работы. Схема подключения.....	38
36.	Инфракрасный датчик расстояния. Принцип работы. Схема подключения.....	39
37.	Сферы применения беспроводных сетей. Преимущества беспроводных сетей.	40

38. Классификация беспроводных технологий коммуникаций	41
39. Диаграмма сетевых телекоммуникационных характеристик BWN.....	43
40. Виды беспроводных сетей. Сети сотовой связи.	44
41. Беспроводные платформы. Методы доступа.....	45
42. Эволюция систем мобильной сотовой связи. 1946 – 1991.....	46
43. Эволюция систем мобильной сотовой связи. 1991 – 2007+.....	47
44. Эволюция систем высокоскоростного доступа. Перечень рабочих групп IEEE 802.	48
45. ОС реального времени. Признаки систем жесткого и мягкого реального времени.....	49
46. Определение операционной системы реального времени. Требования. Критические ресурсы.....	50
47. Требования, накладываемые на вычислительную установку реального времени.....	51
48. Области применения систем реального времени.	52
49. Вычислительные установки, на которых применяются СРВ.	53
50. ОСРВ с монолитной архитектурой.....	54
51. ОСРВ на основе ядра.	55
52. Объектно-ориентированная ОСРВ.....	56
53. ОС реального времени FreeRTOS. Основные характеристики. Применения.....	57
54. ОС реального времени QNX. Основные характеристики. Применения.	58
55. Мультипрограммирование. Характерные показатели эффективности вычислительных систем.	59
56. Подсистема управления процессами и потоками. Создание процесса. ...	60
57. Переход от выполнения одного потока к другому. Завершение процесса.	61
58. Основные данные в контексте процесса. Состояние процесса. Очереди процессов.	62
59. Реализация пакета потоков (ULT). Преимущества. Недостатки.....	63
60. Реализация пакета потоков (KLT). Преимущества. Недостатки.....	64

1. Классификация и структура микроконтроллеров.

В настоящее время выпускаются микроконтроллеры следующих типов:

- 1) 8-разрядные для встраиваемых приложений (Arduino) - Motorola, Intel, Atmel
- 2) 16-разрядные и 32-разрядные микроконтроллеры
- 3) Цифровые сигнальные процессоры (DSP)

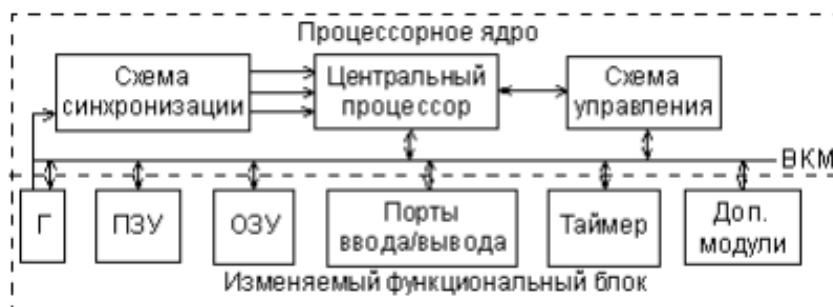


Рис. 27.1. Модульная организация МК.

Структура МК:

- 1) Процессорное ядро
- 2) Изменяемый функциональный блок

Процессорное ядро:

- 1) Центрального процессора
- 2) Внутренней контрольной магистрали - шины
- 3) Схемы синхронизации МК
- 4) Схемы управления режимами работы МК
- 5) Модули обработки прерываний

Изменяемый функциональный блок:

- 1) Модули памяти
- 2) Порты ввода-вывода
- 3) ШИМ
- 4) Таймеры
- 5) Дополнительно – АЦП и ЦАП, компараторы напряжения

2. Микроконтроллер. Основные классификационные признаки МК.

МК – это производные от микропроцессоров интегральные микросхемы с ЦПУ, памятью и устройством ввода/вывода. В микроконтроллере законченная система обработки данных использует только то, что дает ей МК. МК имеет строгий набор задач и простые инструкции.

Современные 8-разрядные МК обладают признаками:

- 1) Модульная организация, при которой на базе одного процессорного ядра (ЦП) проектируется ряд (линейка) МК, различающихся объемом и типом памяти программ, объемом памяти данных, набором периферийных модулей
- 2) Использование закрытой архитектуры. Микроконтроллер – законченная система обработки данных, наращивание возможностей которой не предполагается
- 3) Использование типовых функциональных периферийных модулей
- 4) Расширение числа режимов работы периферийных модулей

Классификация

Разрядность – определяется шириной регистров данных.

- 1) 4-разрядные
- 2) 8-разрядные – блоки по 8 бит
- 3) 16-разрядные и 32-разрядные
- 4) Цифровые сигнальные процессоры DSP в системах обработки сигналов

Архитектура с точки зрения системы команд

- 1) CISC - Complicated Instruction Set Computer – реализация полной системы команд
- 2) RISC - Reduced Instruction Set Computer – реализация сокращенной системы команд

3. Тактовая частота. Производительность. Потребляемая мощность.

Тактовый сигнал – это сигнал, который используется для согласования операций в цифровых схемах. В контексте МК тактовый сигнал используется для согласования операций внутри МК

Такт процессора или такт ядра процессора - промежуток между двумя импульсами тактового генератора, который синхронизирует выполнение всех операций процессора.

Тактовая частота - частота синхронизирующих импульсов синхронной электронной схемы, то есть количество синхронизирующих тактов, поступающих извне на вход схемы за одну секунду. Тактовая частота - это количество тактовых сигналов за одну секунду.

Тактовый генератор – генерирует импульсы на определенной частоте для синхронизации.

Способы тактирования МК - RC-генератор импульсов, Внешний кварц, Внешний генератор и RC-цепочка

Производительность – способность МК выполнять операции вычисления и обрабатывать их с определенной скоростью, она обуславливается:

- 1) Системой памяти программ
- 2) Тактовой частотой, скоростью шины
- 3) Аппаратурой
- 4) Оптимизацией программного обеспечения, выполняемого на МК

Потребляемая мощность – это электрическая мощность, которую потребляет МК при работе. Это энергия, которую МК использует для выполнения вычислений, обуславливается:

- 1) Архитектурой
- 2) Частотой тактового сигнала
- 3) Нагрузкой процессора
- 4) Режимami энергосбережения
- 5) Использованием периферии

4. Сферы применения микроконтроллеров.

МК содержит помимо центрального процессора (ЦП) память и многочисленные устройства ввода-вывода – АЦП, таймеры, ШИМ. Назначение – использование в системах автоматического управления, встроенных в разные устройства – стиральные машины, микроволновые печи, охранные сигнализации, видеокамеры, музыкальные центры. В таких системах микроконтроллер управляет электронными устройствами и осуществляет взаимодействие между ними в соответствии с загруженной в него программой

Встроенные системы управления – системы, у которых основной функцией является функция управления, реализованная на микроконтроллерах. В часах, в микроволновке, в холодильниках, в автомобилях.

Микроконтроллеры можно сопрягать с различными видами датчиковой аппаратуры, после чего отображать информацию, полученную от них, на дисплее, светодиодных индикаторах, ЖК-индикаторах. Также микроконтроллеры могут принимать определенные решения при анализе полученных данных, и отправлять управляющие команды на различные подвластные ей системы

5. Аппаратная платформа Arduino. Характеристики.

Arduino – аппаратная платформа на базе микроконтроллера ATmega328. Платформа имеет 14 цифровых входов-выходов (6 могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, разъем ICSP, кнопку перезагрузки. Для работы необходимо подключить платформу к ПК посредством USB кабеля либо подать питание при помощи адаптера AC/DC или батареи.

Через USB-порт можно подавать питание или загружать прошивки на аппаратную платформу

Питание:

- 1) USB 5V
- 2) Специальный разъем 7V – 12V
- 3) Вход VIN для подачи питания на плату

Рабочее напряжение 5V, рекомендуемое напряжение питания 7-12V, цифровые входы и выходы 0-13, ~ - это ШИМ. Принимают значения LOW и HIGH

Имеет 6 аналоговых входов A0-A5 – они нужны для чтения аналоговых сигналов, то есть непрерывных переменных значений с аналоговых датчиков. При подключении к компьютеру Arduino Uno определяется как виртуальный COM-порт. Светодиоды RX и TX мигают при обмене данными с ПК.

Память:

- 1) 32 КБ Flash-памяти (для прошивок), 2 КБ отведено под bootloader для прошивки Arduino с обычного компьютера через USB
- 2) 1 КБ EEPROM-памяти электрически можно стирать и перезаписывать, хранит данные даже при отключении питания.
- 3) 2 КБ SRAM-памяти (энергозависимая оперативная), которые используются для хранения временных данных. Очищается при обесточивании платформы.

6. Закон Мура. График.

Закон Мура – каждые 2 года (24 месяца) количество транзисторов в процессорах увеличивается в 2 раза. Это эмпирическое наблюдение Гордона Мура

Транзистор — радиоэлектронный компонент из полупроводникового материала, простейший прибор, который по сути похож на кран или электрические ворота. Через транзистор идёт какой-то один ток, а другим током этот поток можно либо пропустить, либо заблокировать

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World
in Data

Transistor count

50,000,000,000

10,000,000,000

5,000,000,000

1,000,000,000

500,000,000

100,000,000

50,000,000

10,000,000

5,000,000

1,000,000

500,000

100,000

50,000

10,000

5,000

1,000

1970 1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020

Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

График, отражающий закон Мура, представляет собой экспоненциальный рост количества транзисторов на интегральной схеме со временем. Ось X представляет собой количество транзисторов, а ось Y – это временные промежутки. График свидетельствует о экспоненциальном росте мощностей

7. Отличительные признаки МК.

Отличительные признаки микроконтроллеров

- 1) Модульная организация, при которой на базе одного процессорного ядра (ЦП) проектируется ряд (линейка) МК, различающихся объемом и типом памяти программ, объемом памяти данных, частотой синхронизации и набором периферийных модулей
- 2) Использование закрытой архитектуры. МК – законченная система обработки данных, наращивание возможностей которой с использованием параллельных магистралей адреса и данных не предполагается. МК имеет строгий набор задач и простые инструкции
- 3) Использование типовых функциональных периферийных модулей (таймеры, процессоры событий, контроллеры последовательных интерфейсов, АЦП)
- 4) Расширение числа режимов работы периферийных модулей, которые задаются в процессе инициализации регистров специальных функций МК

8. Модульная организация МК. Описание.

Модульная организация МК – организация МК, при которой на базе одного процессорного ядра (ЦП) проектируется ряд (линейка) МК, различающихся объемом и типом памяти программ, объемом памяти данных, набором периферийных модулей и частотой синхронизации

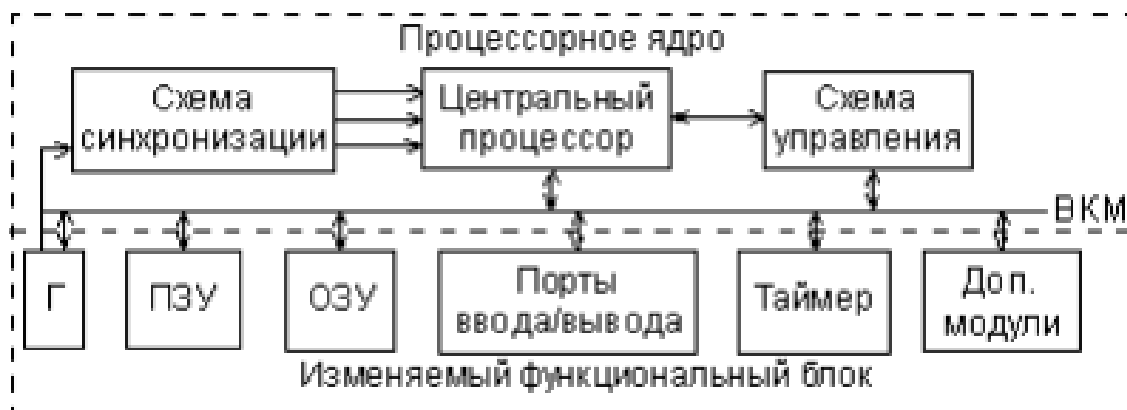


Рис. 27.1. Модульная организация МК.

Процессорное ядро: Центральный процессор (ЦП). Внутренняя коллекторная магистраль (ВКМ) – шины. Схема синхронизации МК. Схема управления режимами работы МК

Изменяемый функциональный блок: Модули памяти разного типа и объема. Порты ввода-вывода. Модули тактовых генераторов. Таймеры. Дополнительные модули – компараторы напряжения, АЦП

9. Память программ. Память данных.

В МК используются 2 основных вида памяти:

1) Память программ – постоянная память (ПЗУ), предназначенная для хранения программного кода (команд) и констант. Ее содержимое в ходе выполнения программы не изменяется. Это энергонезависимая память

Типы: ПЗУ масочного типа заносится при изготовлении МК и не изменяется, ПЗУ EPROM программируемое пользователем с ультрафиолетовым стиранием, ПЗУ OTPROM однократно программируемое пользователем, EEPROM программируемое пользователем с электрическим стиранием и стирание происходит отдельно для каждой ячейки, ПЗУ типа Flash – стирание целыми блоками. Если необходимо изменить содержимое одной ячейки Flash-памяти, потребуется перепрограммировать весь блок

2) Память данных – это память для хранения переменных в процессе выполнения программы (статическое ОЗУ). Это энергозависимая память, не сохраняются при отсутствии питания

Адресные пространства указанных видов памяти, как правило, разделены. Способы адресации и доступа к этим областям памяти также различны. Такая структура позволяет центральному процессору работать одновременно как с памятью программ, так и с памятью данных, что существенно увеличивает производительность

10. Регистровая память. Энергозависимая память. Оперативная память.

МК имеют набор регистров, которые используются для управления его ресурсами и периферийными устройствами. В число этих регистров входят обычно регистры процессора (аккумулятор, регистры состояния, индексные регистры), регистры управления (регистры управления прерываниями, таймером), регистры, обеспечивающие ввод/вывод данных (регистры данных портов, регистры управления параллельным, последовательным или аналоговым вводом/выводом)

1) Регистры МК – этот вид памяти включает в себя внутренние регистры процессора и регистры, которые служат для управления периферийными устройствами – регистры – процессора, управления и ввода-вывода

Регистр – ячейка памяти, которая находится внутри МК и хранит данные для управления работой МК. В регистрах хранятся операнды, результаты вычислений

2) Энергозависимая память – память, которая не способна хранить данные при отсутствии электрического питания

3) Оперативная память – это память, имеющая байтовый формат и являющаяся энергозависимой частью системы памяти. Используется для кратковременного хранения данных. Здесь хранятся выходные и промежуточные данные. ОЗУ – оперативно-запоминающее устройство, реализующее функции оперативной памяти по временному хранению данных (промежуточные вычисления). Данные, хранящиеся в ОЗУ, могут быть сохранены только при подаче питания на МК

11. Периферия МК. Описание.

Периферия МК – это внешняя часть, удаленная от МК, устройства, которые не являются основными. В МК все части системы находятся внутри одного корпуса, но понятие периферийные используется по отношению к ЦП.

Набор периферийных устройств:

- 1) Таймеры и счетчики – такие устройства или модули в МК, которые считают до определенной величины или до такой, какие они по битности. Они считают со скоростью тактовой частоты МК. Это задачи отсчета времени и планирования действий, генерации сигналов, измерение параметров сигналов относительно времени.

Формирование временных интервалов. Подсчет внешних импульсов на выводах.

Формирование последовательных импульсов. Формирование ШИМ

- 2) Тактовые генераторы

- 3) Генераторы ШИМ широтно-импульсной модуляции

МК обычно не могут выдавать произвольное напряжение. Они могут выдать либо напряжение питания 5В, либо землю 0В. Но уровнем напряжение управляется многое – яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения используется ШИМ.

Скважность – отношение времени включения и выключения

- 4) Аналоговый компаратор нужен для сравнения двух сигналов.

- 5) АЦП – устройство преобразует входной аналоговый сигнал в цифровой сигнал

Аналого-цифровое преобразование – это процесс преобразования входной физической величины в ее числовое представление (в цифровой двоичный код).

Аналого-цифровой преобразователь

- 6) Последовательный интерфейс SPI для обмена данными между МК и периферией

- 7) RTC-система реального времени

12. Цифровые порты ввода/вывода.

Каждый МК имеет некоторое количество линий ввода/вывода, которые объединены в многоразрядные параллельные порты ввода/вывода. В памяти МК каждому порту ввода/вывода соответствует свой адрес регистра данных. Типы параллельных портов: однонаправленные порты, двунаправленные порты, порты с альтернативной функцией (мультиплексированные порты), порты с программно управляемой схемотехникой входного/выходного буфера.

Порт – физическое или логическое соединение, через которое принимаются и отправляются данные. Цифровые порты ввода-вывода (I/O ports) - это устройство, предоставляющие процессору микроконтроллера доступ к внешнему миру, посредством получения и формирования цифрового сигнала, то есть дающий возможность нам управлять/общаться с другими цифровыми устройствами. Контакт порта ввода-вывода одновременно может находиться в одном из двух общих состояния – на чтение и на запись. Цифровые порты работают с цифровыми сигналами – 0 и 1.

Типы цифровых входов:

1) Простой цифровой вход. Два диода, расположенные в микроконтроллере, защищают вход от слишком большого напряжения или напряжения отрицательной полярности

2) Вход с триггером Шмитта. Промежуточный триггер не оказывает влияния на работу входа, с точки зрения логической функции. Но он обеспечивает более четкую фиксацию уровней напряжения соответствующих логическим состояниям

3) Вход с подтягивающим резистором. Подтягивающие резисторы на входах цифровых элементов устанавливаются для исключения их неопределенного состояния

Типы цифровых выходов:

1) Стандартный двухконтактный выход. Есть защитные диоды. Два транзистора, которые никогда не бывают открытыми одновременно, обеспечивают формирование на выходе и уровня логической единицы (открыт верхний транзистор) и уровня логического нуля (открыт нижний транзистор)

2) Открытый коллектор, открытый сток. Нет верхнего транзистора, это исключает возможность формировать вытекающий ток. То есть, такой выход может обеспечить только состояние логического нуля, а вместо логической единицы вывод оказывается в подвешенном состоянии

Количество цифровых портов в: Arduino Uno – 14, Arduino Nano – 14, Arduino Mega – 54. Функции Arduino следующие: pinMode(), digitalRead(), digitalWrite()

13. Входные элементы управления. Схемы подключения.

Входные элементы управления - это компоненты или устройства, которые используются для ввода сигналов или команд в систему управления

1) Тумблеры. Нужны для коммутации цепей управления

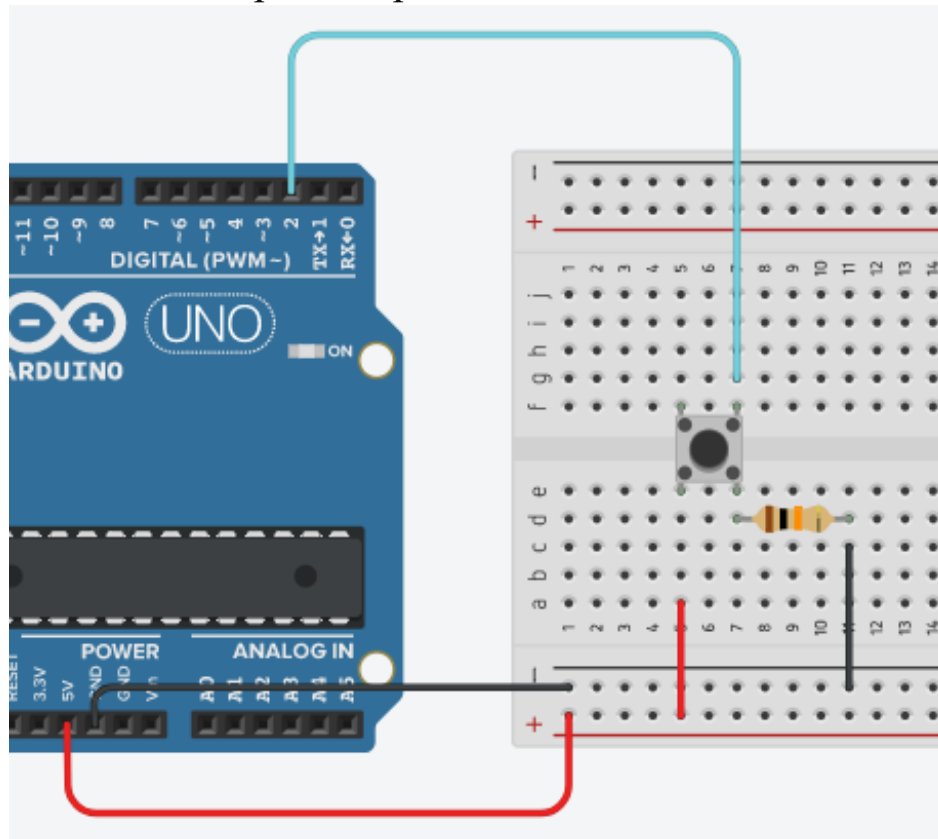
2) Кнопка – механизм, замыкающий цепь, пока есть давление на толкатель.

Дребезг контактов – кнопка замыкается не сразу, какое-то время она дребезжит, «ложно срабатывает»

Подключение кнопки:

1) Кнопка в режиме INPUT_PULLUP Функция `pinMode(PIN_BUTTON, INPUT_PULLUP)` и `digitalWrite(PIN_BUTTON, HIGH)`. Микроконтроллер имеет “встроенную” подтяжку ног к VCC, что даёт возможность подключать кнопку только к GND и пину, но режим работы пина нужно выбрать INPUT_PULLUP. Просто подключить 1 ногу к пину, другую ногу к GND

2) Кнопка с подтягивающим резистором



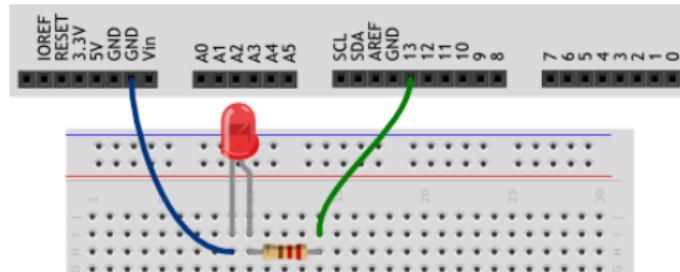
3) Потенциометр – это электрическое устройство, значение уровня сопротивления которого можно задать в определенных пределах. Черный правый – земля, красный левый – питание 5V, средний центр – аналоговый вход (A0 – A5). Считать показания потенциометра – это функция `analogRead(пин)`

4) Датчики – фоторезистор, термистор. Просто подключить 1 пин к аналоговому пину, другой к земле. Полярность не имеет значения

5) Джойстик – пин земли, пин питания, пин кнопки на цифровой пин, пин ось X и пин ось Y на аналоговые пины

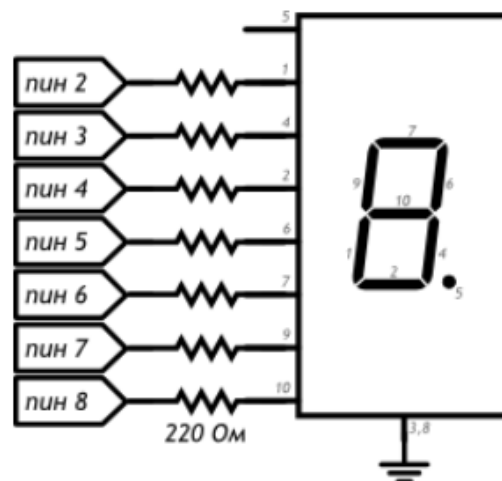
14. Выходные элементы управления. Схемы подключения.

- 1) Элементы индикации – единичные светодиоды или светодиодные сборки, 7-сегментные индикаторы. Светодиод - это полупроводниковый элемент, который, при прохождении через него электрического тока излучает свет. Ток проходит от анода (+) к катоду (-). + - это длинная нога, - - это короткая нога



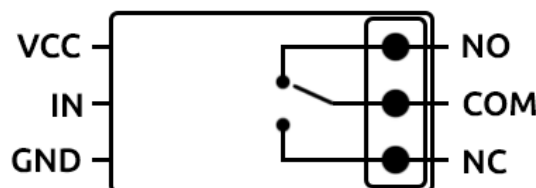
RGB-светодиод – светодиод, способный излучать свет разных цветов. Фактически это три разных светодиода под одной линзой с 1 общим контактом. Имеет входы R, G, B и GND

7-сегментный индикатор – небольшой LED-индикатор, в котором находится 7 светодиодов и 8 в виде точки



Светодиодная шкала – десяток отдельных светодиодов каждый с анодом и катодом. Нужны еще резисторы для каждого диода

- 2) Реле – позволяет подключить устройства, работающие в режимах с большими токами или напряжением. Мы не можем напрямую подключить к плате Arduino мощные насосы, двигатели, даже обычную лампочку накаливания – плата не предназначена для такой нагрузки и работать не будет. Именно поэтому придется добавить реле



VCC – питание, GND – земля, IN – логический управляющий сигнал. COM – общий контакт. NO – нормально разомкнутый контакт относительно COM контакт. NC – нормально замкнутый относительно COM контакт

15. Прерывания. Таймеры/счётчики.

Прерывание – внутреннее или внешнее событие, прерывающее обработку текущей программы и переходящее к некоторой другой. Это механизм реакции на внешние события. Этот механизм работает таким образом, что при наступлении некоторого события в процессоре возникает сигнал, заставляющий процессор прервать выполнение текущей программы

- Одноуровневые прерывания – при прерывании процессор аппаратно переходит к подпрограмме обработки прерывания по определенному адресу. Для обработки всех прерываний используется только 1 точка входа

- Векторные прерывания – многоуровневые прерывания. В векторных прерываниях каждому источнику прерывания соответствует свой адрес

- Приоритетные прерывания

- Вложенные прерывания - При вложенных прерываниях, процедура обработки текущего прерывания может быть прервана (отложена) при поступлении запроса на прерывание, имеющего более высокий уровень приоритета. После обработки прерывания с более высоким уровнем приоритета процессор возвращается к прерванной процедуре и продолжает обработку данного прерывания до ее окончания или до нового прерывания

- Внешние – событие при сигнале на входе МК

- Внутреннее – работа периферии – АЦП, таймер

Таймеры и счетчики – устройства или модули в МК, которые считают до определенной величины или до такой величины, какой они битности. Считают они постоянно с одной скоростью, со скоростью тактовой частоты микроконтроллера. Таймер микроконтроллера – это цифровой счётчик, осуществляющий подсчёт количества подаваемых на него импульсов

- 1) Формирование временных интервалов

- 2) Подсчет внешних импульсов на выводах

- 3) Формирование последовательных импульсов

- 4) Формирование ШИМ с программируемой частотой и скважностью

У сторожевого таймера только одна задача – производить сброс МК

16. Работа с аналоговыми сигналами. АЦП.

АЦП (аналогово-цифровой преобразователь) позволяет превращать аналоговый сигнал в цифровой код, пригодный для использования в ЭВМ. На карте Arduino Uno аналоговые входы имеют буквенно-цифровые обозначения A0, A1 ... A5. Для считывания аналогового сигнала используется функция `analogRead(номер_контакта)`. Аналоговые пины очень часто используются при работе с потенциометрами (переменный резистор). При помощи полученного значения можно влиять на ход работы программы, менять какие-то настройки и тому подобное. У потенциометра всегда три ноги: две крайние и одна центральная. К Arduino потенциометр подключается следующим образом: средний вывод на любой А-пин, крайние – на GND и питание.

Ввод аналогового сигнала функция `analogRead(пин)`, вывода `analogWrite(номер_контакта, уровень_сигнала)` только на ШИМ с волной пины на Ардуино – 3, 5, 6, 9, 10 и 11

АЦП – устройство преобразует входной аналоговый сигнал в цифровой сигнал (дискретный код)

Аналого-цифровое преобразование – это процесс преобразования входной физической величины в ее числовое представление (в цифровой двоичный код). Аналого-цифровой преобразователь – устройство, выполняющее такое преобразование

- Аналоговый сигнал подается на АЦП от датчика или сенсора
- Аналоговый сигнал разбивается на дискретные значения
- Аналоговый диапазон делится на конкретное число уровней. Число уровней определяется разрядностью АЦП
- В регистре АЦП сохраняются цифровые значения и далее используются в МК путем чтения их из регистров АЦП

Характеристики АЦП – частота преобразования (отсчеты в секунду) и разрядность (бит)

17. Тактовый генератор. Система реального времени.

Тактовый генератор – автогенератор, формирующий рабочие такты процессора (частоту). Кроме тактирования ЦП в обязанности тактового генератора входит еще организация циклов системной шины. Тактовый генератор генерирует периодический сигнал – тактовый сигнал (импульс). Этот сигнал имеет фиксированную частоту и определяет скорость, с которой данные обрабатываются и передаются на МК. По сигналу тактового генератора происходят операции записи/чтения регистров и памяти и все операции

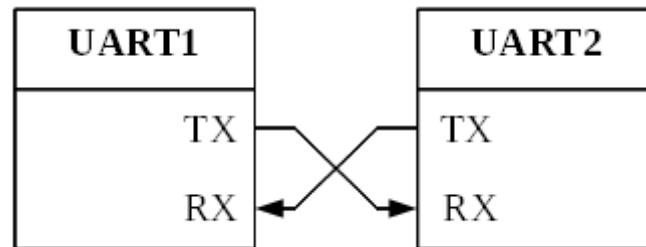
Работа МК зависит от их источников тактовых импульсов. Процессор, шина и периферийные устройства используют тактовый генератор для синхронизации операций. Тактовый генератор определяет, насколько быстро процессор выполняет инструкции

Система реального времени реагирует за предсказуемое время на непредсказуемые события. Операции должны быть выполнены строго в определенных временных предсказуемых рамках. Такая система требует точной синхронизации и управления временем для обработки команд в реальном времени. Система реального времени необходима, когда при наличии нескольких процессов и устройств время выполнения процессов важнее, чем средняя производительность. То есть когда стоит задача запуска нескольких процессов в определенное время. Система гарантирует тайминг выполнения требуемых задач, имеет меньшую задержку, может точно определить, выполнена ли задача, и дает возможность выполнять как срочные, так и некритичные по времени задачи. Главное – задачи в срок. В такой системе тактовый генератор обеспечивает основной временной сигнал, который синхронизирует все операции и устройства. Он определяет базовую тактовую частоту, по которой происходят вычисления, обмен данными на МК.

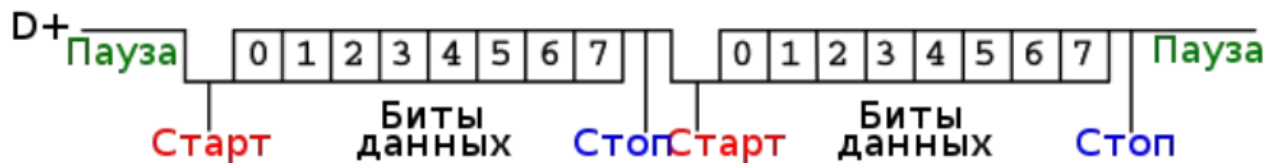
18. Последовательный интерфейс UART/USART. Подключение. Формат данных.

UART – универсальный асинхронный приемопередатчик – Universal Asynchronous Receiver-Transmitter. Универсальный – распространенный. Асинхронный – прием и передача отдельных битов не выравниваются синхросигналами.

Два пина контроллера – RX и TX (Receiver и Transmitter) – подключение:



Данные передаются в инверсном коде (логическая 1 – низкий уровень, логический 0 – высокий уровень). Формат передачи данных - в UART передача данных происходит в последовательной форме, т.е. по одному биту



- 1) Стартовый бит всегда логический 0
- 2) Передача битов данных от 5 до 9 (обычно 8 бит)
- 3) Посылка завершается стоп-битами логическая 1, количество один, полтора или два
- 4) Начинается передача следующей посылки или пауза произвольной длительности (длинная логическая 1)

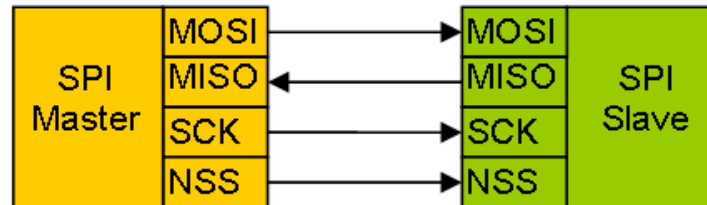
Класс Serial в Arduino для UART: `Serial.begin(long)`: Запускает порт с указанной скоростью. `Serial.end()`: Останавливает порт, если он был запущен. `Serial.available()`: Возвращает количество принятых байтов в буфере порта. `Serial.read()`: Возвращает байт из буфера приема или 0xFFFF, если буфер пуст. `Serial.print(xxx)`: Передает данные в порт (байты, строки, числа). `Serial.println(xxx)`: Передает данные с переносом строки. `Serial.write(xxx)`: Передает двоичные данные в порт и возвращает число переданных байтов.

USART – синхронно асинхронный приемопередатчик – это UART с синхронизацией сигналов. Это полнодуплексный обменный интерфейс для общения двух устройств. UART – это USART без синхронизации.

РАЗНИЦА - Сигналы тактирования UART генерируются внутри МК и синхронизируются с потоком данных по переходу стартового бита. Каких-либо входящих синхросигналов, связанных с данными, здесь нет. Напротив, USART может быть настроен для работы в синхронном режиме. В этом режиме отправляющая данные периферия генерирует сигнал синхронизации, который периферией приемной стороны может быть извлечен из потока данных без априорной информации о скорости передачи

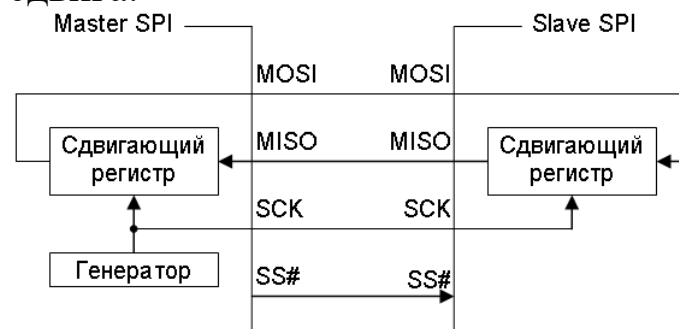
19. Последовательный интерфейс SPI. Подключение. Формат данных.

SPI сериальный периферийный интерфейс, последовательный интерфейс периферийных устройств - интерфейс, обеспечивающий полнодуплексную синхронную последовательную связь между устройствами; для связи используются четыре однонаправленные сигнальные линии. Мастер – например МК, слейв – периферийное устройство



- 1) MOSI Master Output, Slave Input – от МК на периферию
- 2) MISO Master In / Slave Out – с периферии на МК
- 3) SCK Serial Clock – тактовый сигнал, генерируется ведущим устройством и управляет ведомым
- 4) NSS Slave Select – выбор ведомого устройства

Используется регистр сдвига.



- 1) Данные передаются со старшего бита. Передача пакетами. Пакет 1 байт (8 бит)
- 2) В сдвиговые регистры данные помещаются с младшего бита
- 3) Каждый обмен сопровождается отправкой и получением данных одинакового объема
- 4) Обмен инициируется МК (ведущее устройство), выбирает нужное ведомое устройство сигналом SS#
- 5) SPI – всегда взаимный обмен данными
- 6) Чтобы МК начал передавать данные, надо записать в его регистр данных. Он загружает из регистра данных в сдвигающий регистр эти данные, начинает формировать тактовый сигнал, сдвигающий регистр по 1 биту передает данные
- 7) Периферия работает по внешнему тактовому сигналу от МК и не может само управлять процессом пересылки!

+ : полнодуплекс, произвольный выбор длины пакета (не ограничен 8 битами), простота аппаратной реализации, только четыре вывода, высокая пропускная способность и полнодуплексная передача данных - : ведомое устройство не управляет потоком, нет подтверждения приема данных со стороны ведомого, нет официального стандарта, есть множество реализаций интерфейса

20. Двухпроводной последовательный интерфейс TWI/I2C. Подключение. Формат данных.

Интерфейс для обмена данными между МК, датчиками и исполнительными устройствами. Это протокол синхронной связи – обмен данными происходит по общему для всех связанных устройств сигналу синхронизации. I2C использует 2 двуправленные линии связи – SDA и SCL. SDA – линия данных. Ею управляют как Master, так и Slave, в зависимости от направления передачи. SCL – линия тактирования. Master – ведущий, обычно МК. Slave – ведомые, периферия.

Единица обмена информацией – пакет, обрамленный стартовым и стоповым битами. Мастер в начале каждого пакета передает 1 байт, где указывает адрес ведомого и направление передачи данных. Данные – как 8битные слова. После каждого слова передается один бит подтверждения приема

Процедура обмена - ведущий формирует состояние СТАРТ: при ВЫСОКОМ уровне на линии SCL он генерирует переход сигнала линии SDA из ВЫСОКОГО состояния в НИЗКОЕ. Этот переход воспринимается всеми устройствами, подключенными к шине, как признак начала процедуры обмена. Процедура обмена завершается тем, что ведущий формирует состояние СТОП – переход состояния линии SDA из НИЗКОГО состояния в ВЫСОКОЕ при ВЫСОКОМ состоянии линии SCL. Состояния СТАРТ и СТОП всегда вырабатываются ведущим. Считается, что шина занята после фиксации состояния СТАРТ. Шина считается освободившейся через некоторое время после фиксации состояния СТОП

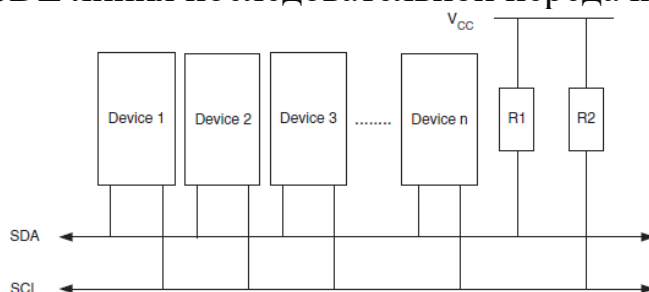
Для корректной передачи данных, приемник должен подтверждать прием каждого байта от передатчика. Для этого в спецификации протокола обмена по шине I2C вводится специальный бит подтверждения, выставляемый приёмником на шину SDA после приема каждого 8-го бита данных.

Структура сообщения на первый взгляд выглядит намного сложнее, чем у старого доброго UART.



I2C является авторской разработкой компании Phillips и поэтому Atmel, чтобы не нарушать авторские права, немного изменила наименование данной шины и назвала её TWI (Two-wire Serial Interface)

Описание TWI - Двухпроводной последовательный интерфейс TWI (Two-wire Serial Interface) является полным аналогом базовой версии интерфейса I2C (двухпроводная двуправленная шина) фирмы Philips. Устройства, поддерживающие I2C, обычно также поддерживают TWI, поскольку TWI является совместимым вариантом I2C. И I2C, и TWI обеспечивают двухпроводной интерфейс обмена данными. 2 линии - SCL линия синхронизации и SDA линия последовательной передачи данных



21. Интерфейс JTAG. Подключение. Формат данных.

Интерфейс JTAG (Joint Test Action Group) представляет собой синхронный интерфейс, который используется для тестирования и программирования интегральных микросхем. Он состоит из нескольких логических устройств, включая TAP (Test Access Port) контроллер, который управляет передачей данных между внешним устройством и цепями внутри микросхемы. Сигналы в интерфейсе JTAG принимаются по переднему фронту синхроимпульсов, что означает, что данные считываются по фронту сигнала тактового импульса. Младшие биты данных передаются первыми, а старшие биты - последними. Передача данных происходит только в течение состояний TAP-контроллера, таких как Shift-DR (Shift Data Register) и Shift-IR (Shift Instruction Register). В этих состояниях происходит сдвиг данных через регистры для передачи или приема информации.

Выходные данные выдвигаются по заднему фронту синхроимпульсов, что означает, что данные выводятся на выход только после завершения передачи всего пакета данных.

Подключение к устройству с помощью интерфейса JTAG осуществляется через специальный JTAG-коннектор, который обычно имеет 4 или более контакта. Этот коннектор обеспечивает удобное и надежное соединение между устройством и программатором или отладчиком: определение пинов JTAG, подключение к программатору или отладчику, подключение проводов, настройка программного обеспечения, выполнение операций.

Формат данных, передаваемых по интерфейсу JTAG, зависит от конкретного протокола. Наиболее распространенным протоколом является Boundary-Scan, который позволяет проводить тестирование цепей на плате и программирование микросхем.

Для отправки команд и данных по интерфейсу JTAG используются специальные инструкции, которые управляют работой устройства и обменом информацией. Формат данных может включать в себя команды, адреса, данные для чтения или записи, а также специальные биты для управления процессом тестирования или программирования.

При подключении интерфейса JTAG к устройству данные передаются в формате последовательных битов. Этот формат называется TAP (Test Access Port) Controller State Machine и состоит из нескольких состояний, включая:

1. Test-Logic-Reset (TLR): В этом состоянии все устройства в цепи JTAG сбрасываются в начальное состояние и готовы к передаче данных.

2. Run-Test/Idle (RTI): В этом состоянии устройства находятся в режиме ожидания и готовы к выполнению операций.

3. Select-DR-Scan (SDR): В этом состоянии выбирается регистр данных (Data Register), в который будут передаваться данные.

4. Capture-DR (CDR): В этом состоянии данные с линий TDI (Test Data In) передаются в выбранный регистр данных.

5. Shift-DR (SDR): В этом состоянии данные с линии TDI сдвигаются внутри регистра данных.

6. Exit1-DR (EDR) и Exit2-DR (EDR): В этих состояниях данные готовы к обработке в устройстве.

7. Update-DR (UDR): В этом состоянии данные из регистра данных обновляются в устройстве.

22. Интерфейс CAN. Подключение. Формат данных.

Интерфейс CAN (Controller Area Network) – это широко используемый серийный коммуникационный протокол, который обеспечивает надежную и эффективную передачу данных между устройствами в реальном времени. При подключении устройств через интерфейс CAN используется двухпроводная линия для обмена данными. Эта двухпроводная линия состоит из двух сигнальных проводов: CAN-High (CAN-H) и CAN-Low (CAN-L).

Вот как происходит обмен данными по CAN интерфейсу через двухпроводную линию:

1. данные передаются как разность потенциалов между CAN-H и CAN-L.
2. данные передаются в форме кадров, которые содержат заголовок, идентификатор сообщения, данные и CRC (циклический избыточный код).
3. может использоваться мастер-слейв архитектура, где одно устройство (мастер) инициирует обмен данными, а другие устройства (слейвы) отвечают на запросы или передают информацию.
4. поддерживает различные скорости передачи данных, от низких (например, 125 kbps) до высоких (например, 1 Mbps).

Формат данных в CAN-сообщениях следующий:

1. ID (Идентификатор): каждое CAN-сообщение содержит уникальный идентификатор, который определяет приоритет сообщения и его содержание. ID может быть 11-битным (стандартный формат) или 29-битным (расширенный формат).
2. DLC (Длина данных): DLC указывает на количество байт данных, передаваемых в сообщении. Может быть от 0 до 8 байт.
3. Data (Данные): Это фактические данные, передаваемые в сообщении.
4. CRC (Контрольная сумма): Контрольная сумма используется для обеспечения целостности данных и проверки ошибок при передаче.
5. ACK (Подтверждение): Устройства в сети CAN могут подтверждать успешное получение сообщения.

23. Языки программирования МК. Особенности.

Наиболее популярные языки программирования для микроконтроллеров:

- 1) **Assembler:** яп низкого уровня, который позволяет программисту точно управлять ресурсами микроконтроллера. Assembler позволяет максимально эффективно использовать ресурсы микроконтроллера, но требует более глубокого понимания аппаратной части. Он позволяет программисту работать непосредственно с аппаратными ресурсами микроконтроллера: имеется прямой доступ к регистрам процессора, памяти и другим аппаратным ресурсам микроконтроллера, позволяет достичь максимальной производительности микроконтроллера, программы обычно имеют меньший объем кода, можно точно управлять обращениями к памяти.
- 2) **C:** обеспечивает баланс между производительностью и удобством программирования: код на C легко переносится между различными микроконтроллерами и платформами, прямое управление аппаратными ресурсами микроконтроллера, имеет обширную стандартную библиотеку функций, которая упрощает разработку приложений для микроконтроллеров.
- 3) **C++:** поддерживает объектно-ориентированное программирование, что позволяет организовывать код в виде объектов с методами и свойствами. Это упрощает разработку сложных систем на микроконтроллерах и повышает их модульность и переиспользуемость, позволяет использовать шаблоны для создания обобщенных алгоритмов и структур данных, позволяет напрямую работать с памятью и регистрами устройств, поддерживает наследование и полиморфизм, что позволяет создавать иерархии классов и переопределять их методы.
- 4) **MicroPython:** упрощенная версия языка программирования Python, оптимизированная для использования на микроконтроллерах: простой и понятный синтаксис, код выполняется интерпретатором непосредственно на микроконтроллере, имеет небольшой размер, встроенная поддержка для периферийных устройств, поддерживает механизмы для добавления пользовательских модулей и библиотек.
- 5) **Arduino IDE (язык Wiring):** Arduino IDE использует язык программирования Wiring, который является диалектом языка C/C++: простой и понятный синтаксис, обширная библиотека функций, которая упрощает вз-ие с различными периферийными устройствами, предоставляет широкий выбор стандартных и сторонних библиотек, которые расширяют функциональность микроконтроллеров, удобная среда разработки с возможностью загрузки кода на микроконтроллер, мониторинга последовательного порта и отладки программ, совместимость с различными платформами.

24. Типы корпусов МК. Описание. Назначение.

Существует множество типов корпусов микроконтроллеров, каждый из которых имеет свои особенности и применение:

1) DIP (Dual In-line Package): представляет собой прямоугольный пластиковый корпус с двумя рядами контактов, которые выступают сбоку от корпуса, такие корпуса используются для прототипирования в устройствах, где не требуется высокая плотность упаковки.

2) QFP (Quad Flat Package): QFP-корпус имеет плоский корпус с контактами, расположенными по периметру корпуса, обеспечивают более высокую плотность упаковки по сравнению с DIP и широко используются в современных микроконтроллерах.

3) LQFP (Low-profile Quad Flat Package): корпус имеет ту же конструкцию, что и QFP, но с более низким профилем, позволяют уменьшить высоту устройства и используются в компактных приложениях.

4) BGA (Ball Grid Array): корпус имеет шаровые контакты на нижней части корпуса, обеспечивают высокую плотность упаковки и хорошее тепловое соединение, что делает их популярными для микроконтроллеров с высокой производительностью.

5) SMD (Surface Mount Device): могут быть различных форм и размеров, но их общая особенность - это возможность монтажа на поверхность печатной платы, широко используются в современной электронике из-за своей компактности и возможности автоматической сборки.

Анимация – это зависимость свойства графического примитива от его аргументов по определенному правилу. Анимировать можно практически любое свойство примитива (в простейшем случае, например, размер или цвет), настраивая встроенный алгоритм преобразования или используя скрипт.

SCADA DataRate представляет разработчикам полный спектр инструментов, необходимых для создания качественной системы визуализации под разные классы задач.

Система визуализации SCADA DataRate это объектно-ориентированная графическая подсистема с множеством графических примитивов с полным набором операций редактирования (как отдельного примитива, так и структурированного изображения), возможностями анимации свойств объектов, реакций на события и динамической визуализацией.

Графическая подсистема представляет собой совокупность графических объектов – рабочих столов, мнемосхем, графических примитивов, а также изображений из библиотеки **Image Library**. Для каждой мнемосхемы автоматически строится **Дерево примитивов мнемосхемы**, которое содержит все графические элементы и их свойства.

HMI (Human-Machine Interface) – это система, позволяющая взаимодействовать человеку с машиной или устройством. Это может включать в себя элементы аппаратного и программного обеспечения, такие как сенсорные экраны, кнопки, индикаторы, звуковые сигналы и другие элементы.

HMI используется для управления и мониторинга различных устройств и систем, таких как промышленные автоматизированные системы, медицинское оборудование, автомобильные панели управления и другие. Цель HMI - обеспечить удобный и эффективный способ взаимодействия человека с устройством.

При работе с HMI-дисплеями **DWIN среда разработки DGUS** позволяет создавать анимацию с нуля, используя встроенный визуальный редактор для настройки размера, положения и скорости анимации.

При создании анимации сначала необходимо разработать изображение. Это можно сделать в Photoshop, Illustrator или Corel Draw. После того как изображение будет готово, его нужно загрузить на дисплей DWIN HMI, а затем создать анимацию. Для этого используется визуальный редактор DGUS, который позволяет настраивать различные параметры, такие как начальная и конечная точки прокрутки, скорость и количество прокрутки. После создания анимации, её можно протестировать на реальном дисплее и внести необходимые коррективы.

26. Микроконтроллеры STM32. Описание. Характеристики.

Микроконтроллеры STM32 – это семейство микроконтроллеров производства компании STMicroelectronics, которые широко используются во многих электронных устройствах и системах. Они предлагают высокую производительность, надежность и широкий набор периферийных устройств, что делает их популярным выбором для различных приложений, включая управление HMI-дисплеями. Использование среды разработки DGUS с микроконтроллерами STM32 позволяет создавать качественную анимацию для HMI-дисплеев с минимальными затратами времени и усилий.

Описание: STM32 — семейство микроконтроллеров, основанных на 32-битных ядрах ARM Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+ или Cortex-M0 с сокращённым набором инструкций. STMicroelectronics (ST) имеет лицензию на IP-процессоры ARM от ARM Holdings. Дизайн ядра ARM имеет множество настраиваемых опций, и ST выбирает индивидуальную конфигурацию для каждого микроконтроллера, при этом добавляя свои собственные периферийные устройства к ядру микроконтроллера перед преобразованием дизайна в полупроводниковую пластину.

Характеристика: Тактовая частота до 64 МГц, имеют от 32 до 128 кБ флеш памяти, от 8 до 36 кБ ОЗУ. Периферийные модули включают в себя до 60 линий ввода/вывода, 12-битный АЦП (до 16 каналов) и временем выборки 0,4 мкс, до 2-х ЦАП, 7 каналов DMA, до 4-х USART, USART с низким потреблением энергии, до 2-х SPI, до 2-х I2C.

27. Микроконтроллеры AVR. Описание. Характеристики.

Микроконтроллеры AVR (Alf and Vegard's RISC processor) – это семейство микроконтроллеров, разработанных компанией Atmel. Они широко используются во устройствах бытовой техники, автомобильной электроники, промышленных устройств, робототехнике и другие.

Описание: Микроконтроллеры AVR имеют гарвардскую архитектуру (программа и данные находятся в разных адресных пространствах) и систему команд, близкую к идеологии RISC. Процессор AVR имеет 32 8-битных регистра общего назначения, объединённых в регистровый файл. В отличие от «идеального» RISC, регистры не абсолютно ортогональны

Характеристики:

- микроконтроллеры основаны на упрощенной RISC архитектуре, что делает их эффективными и быстрыми в обработке команд.
- обычно имеют встроенный процессор с различными характеристиками, такими как тактовая частота, объем памяти, набор инструкций и т. д.
- обычно имеют встроенную флэш-память для программ и энергонезависимую EEPROM для хранения данных.
- имеют разнообразные периферийные устройства, такие как аналогово-цифровые преобразователи (ADC), таймеры, интерфейсы UART, SPI, I2C и другие.
- обладают низким энергопотреблением, что делает их идеальным выбором для устройств с ограниченным источником питания.
- могут быть запрограммированы с использованием различных сред разработки, таких как Atmel Studio, Arduino IDE и другие.

28. Микроконтроллеры ESP8266. Описание. Характеристики.

Микроконтроллеры ESP8266 – это популярное семейство микроконтроллеров, разработанных компанией Espressif Systems. Они широко используются в различных проектах Интернета вещей (IoT) и беспроводных приложениях благодаря своей встроенной поддержке Wi-Fi. Он представляет собой сетевое решение с Wi-Fi-трансивером на борту плюс возможность выполнения записываемых в его память приложений. Это Wi-Fi устройство, подключиться к нему можно тоже через Wi-Fi

Характеристики:

- основан на архитектуре Xtensa с 32-битным процессором, что обеспечивает высокую производительность и эффективность.
- имеет встроенный модуль Wi-Fi, который позволяет устройствам подключаться к беспроводным сетям и обмениваться данными через Интернет.
- обычно имеет встроенную флэш-память для программ и данные, а также доступ к внешней SPI Flash памяти.
- обладает разнообразными периферийными устройствами, такими как аналогово-цифровые преобразователи (ADC), интерфейсы UART, SPI, I2C, GPIO и другие.
- обладает различными режимами энергосбережения, что позволяет устройствам работать долгое время от батарейного питания.
- можно программировать с использованием Arduino IDE, MicroPython, Lua и других сред разработки.

29. Микроконтроллеры ESP32. Описание. Характеристики.

Микроконтроллеры ESP32 – это семейство микроконтроллеров, разработанных компанией Espressif Systems. Они являются продвинутым вариантом ESP8266 и обладают более широкими возможностями и функциональностью. Это высокоинтегрированный, совмещённый чип (Wi-Fi + Bluetooth), выполненный для решений, требующих минимальных показателей энергопотребления.

ESP32 разработан для носимой электроники и приложений интернета вещей, выполнен в супер-миниатюрном корпусе 6х6 мм, требующий для интеграции около 10-ти внешних компонентов.

Характеристики:

- ESP32 оснащен двухъядерным процессором Xtensa, что обеспечивает более высокую производительность и возможность параллельной обработки задач.

- ESP32 имеет встроенную поддержку Wi-Fi и Bluetooth, что позволяет устройствам подключаться к беспроводным сетям и обмениваться данными не только через Интернет, но и по беспроводному каналу Bluetooth.

- ESP32 обычно имеет более большой объем встроенной флэш-памяти, что обеспечивает возможность хранения большего объема программ и данных.

- ESP32 обладает разнообразными периферийными устройствами, включая ADC, UART, SPI, I2C, GPIO, Ethernet MAC, Touch Sensor, и другие.

- ESP32 имеет различные режимы энергосбережения, что позволяет устройствам работать долгое время от батарейного питания.

- ESP32 можно программировать с использованием Arduino IDE, MicroPython, FreeRTOS и других сред разработки.

Отличительные особенности: двухъядерный процессор, встроенная поддержка Wi-Fi, и Bluetooth, большой объем памяти, продвинутый режим энергосбережения, более широкий набор периферийных устройств.

30. Сервопривод. Описание. Устройство сервопривода.

Сервопривод – это устройство, которое используется для управления положением объекта. Он состоит из двигателя, обратной связи и контроллера. Когда на сервопривод подается управляющий сигнал, контроллер обрабатывает этот сигнал и отправляет команду двигателю, чтобы изменить положение объекта.

Основной принцип работы сервопривода заключается в использовании обратной связи для поддержания заданного положения объекта. Датчики на сервоприводе постоянно мониторят положение объекта и передают информацию обратно в контроллер. Таким образом, контроллер может корректировать команды двигателю, чтобы добиться необходимого положения.

Основные характеристики сервопривода:

1. Крутящий момент - векторная физическая величина, характеризующая действие силы на механический объект, которое может вызвать его вращательное движение.
2. Рабочее напряжение, которое подается на сервоприводы в допустимых пределах.
3. Максимальный угол поворота
4. Скорость поворота – определяется как время, необходимое для того, чтобы вал повернулся на 60° .
5. Размеры и вес

В состав сервопривода входят следующие компоненты:

1. Мотор – обеспечивает вращение оси.
2. Редуктор – без него скорость вращения вала сервопривода была бы слишком большой.
3. Потенциометр - обеспечивает обратную связь и контроль положения вала.
4. Контроллер – отслеживает текущее положение оси по изменению сопротивления потенциометра. Это позволяет организовать отрицательную обратную связь для удержания вала сервопривода в нужном положении.



31. Сервопривод. Преимущества. Недостатки. Схема подключения.

Сервопривод – это механизм, позволяющий устанавливать и фиксировать рабочий орган оборудования в заданных положениях, перемещать его в соответствии с заданной программой. Они могут поддерживать необходимый момент на валу при нулевой скорости вращения вала. Это используется для удержания исполнительного механизма в одном положении под нагрузкой.

Конструкция может существенно различаться в зависимости от назначения, вне зависимости от области применения устройства содержат следующие узлы:

- Передаточный механизм.
- Электродвигатель.
- Датчики положения и скорости вращения вала.
- Частотный преобразователь.
- Контроллер.

Принцип работы: контроллер программируется на определенный режим работы и выдает сигнал на преобразователь частоты. Устройство подает на электродвигатель напряжение необходимой частоты и величины. Силовой агрегат перемещает исполнительный механизм с заданной скоростью и моментом, соответствующим нагрузке. По достижении заданного положения рабочего органа, подается соответствующий сигнал с датчиков положения на контроллер, который останавливает двигатель.

Преимущества:

1. Точность механического привода не зависит от износа деталей, появления люфтов, термических и механических изменений конструктивных элементов.
2. Диагностирование неисправностей происходит моментально за счет обратной связи.
3. Скорость вращения
4. Экономичность – вращение вала у шаговой электрической машины осуществляется при максимально допустимом напряжении питания, чтобы обеспечить максимальный момент.

Недостатки:

1. Сложность системы управления и необходимость реализации ее работы
2. Необходимость контролировать как частоту вращения, так и принимать меры для принудительного затормаживания в нужной точке – это приводит к дополнительным затратам энергии, программных и механических ресурсов.
3. Обязательно используется дополнительный измерительный блок, контролирующий положение рабочего органа.

Подключается к пластине:

Черный провод на GND

Красный провод на 5V

Желтый провод на аналоговый выход (11)

32. Шаговый двигатель. Описание. Принцип работы. Преимущества. Недостатки.

Шаговый двигатель – это вид электрического двигателя, который вращается на определенный угол при каждом импульсе управления. Он состоит из фиксированного статора и подвижного ротора, который делится на определенное количество шагов. Применяются в различных областях, где требуется точное позиционирование и управление вращением.

Принцип работы шагового двигателя основан на смене последовательности включения фаз обмоток для создания вращательного движения, т.е. поэтапной активации обмоток ротора, которые обеспечивают перемещения, то есть шаги ротора или его фиксацию. Вращение статора и ротора возникает за счёт магнитных потоков. Сила вращения шагового двигателя пропорциональна силе магнитного поля, которая соответствует числу витков в обмотке и показателю тока в ней. Шаговый двигатель – устройство, преобразующее электроимпульсы в механическое движение.

Преимущества шаговых двигателей:

1. Простота управления: шаговые двигатели могут быть точно управляемыми и позиционированными с помощью импульсов управления.
2. Отсутствие обратной связи: не требуется обратная связь для контроля положения ротора, что упрощает систему управления.
3. Высокий крутящий момент на низких скоростях: шаговые двигатели обладают высоким крутящим моментом на низких скоростях, что делает их идеальным выбором для приложений, требующих точного позиционирования.
4. Отсутствием в конструкции щёток, что продлевает срок эксплуатации и повышает надёжность.
5. Способны работать в режиме быстрого старта или остановки, а также перезапуска.

Недостатки шаговых двигателей:

1. Требуют импульсов управления: для работы шагового двигателя необходимы постоянные импульсы управления, что может потребовать дополнительных устройств управления.
2. Низкая эффективность на высоких скоростях: на высоких скоростях шаговые двигатели могут потерять шаги из-за инерции ротора, что может привести к потере позиционирования.
3. Имеют низкую эффективность по сравнению с другими типами двигателей, такими как синхронные или асинхронные двигатели.

33. Шаговый двигатель. Преимущества. Недостатки. Схема подключения.

Шаговый двигатель – это вид электрического двигателя, который вращается на определенный угол при каждом импульсе управления. Он состоит из фиксированного статора и подвижного ротора, который делится на определенное количество шагов. Применяются в различных областях, где требуется точное позиционирование и управление вращением.

Преимущества шаговых двигателей:

1. Простота управления: шаговые двигатели могут быть точно управляемыми и позиционированными с помощью импульсов управления.
2. Отсутствие обратной связи: не требуется обратная связь для контроля положения ротора, что упрощает систему управления.
3. Высокий крутящий момент на низких скоростях: шаговые двигатели обладают высоким крутящим моментом на низких скоростях, что делает их идеальным выбором для приложений, требующих точного позиционирования.
4. Отсутствием в конструкции щёток, что продлевает срок эксплуатации и повышает надёжность.
5. Способны работать в режиме быстрого старта или остановки, а также перезапуска.

Недостатки шаговых двигателей:

1. Требуют импульсов управления: для работы шагового двигателя необходимы постоянные импульсы управления, что может потребовать дополнительных устройств управления.
2. Низкая эффективность на высоких скоростях: на высоких скоростях шаговые двигатели могут потерять шаги из-за инерции ротора, что может привести к потере позиционирования.
3. Имеют низкую эффективность по сравнению с другими типами двигателей, такими как синхронные или асинхронные двигатели.

Для подключения шагового двигателя к системе управления, обычно используется специальный драйвер шагового двигателя, который преобразует управляющие сигналы от контроллера в необходимые фазовые сигналы для двигателя.

Основные элементы:

1. Шаговый двигатель.
2. Драйвер шагового двигателя.
3. Источник питания.
4. Микроконтроллер или другое устройство управления.

Принцип подключения:

1. Подключить выводы шагового двигателя к соответствующим выходам драйвера шагового двигателя.
2. Подключить питание к драйверу шагового двигателя и к шаговому двигателю.
3. Подключить управляющие сигналы от микроконтроллера к соответствующим входам драйвера шагового двигателя.



34. Гироскоп. Акселерометр. Описание. Назначение.

Акселерометр – это датчик, который измеряет ускорение объекта в направлении, в котором он установлен. Существуют различные типы акселерометров, такие как пьезоэлектрические, конденсаторные, MEMS и другие.

Назначение: используется для измерения ускорения, детекции вибраций, определения положения объекта в пространстве, контроля движения и вибрации в различных устройствах, таких как мобильные телефоны, автомобили, дроны, спортивные трекеры и другие.

Акселерометр измеряет проекцию результирующего кажущегося ускорения и позволяет определить угол наклона устройства по отношению к земле, а программа может на основе полученных данных повернуть, например, изображение на экране устройства, в котором этот акселерометр установлен.

Акселерометр в простейшем виде — это закрепленный на пружинке грузик, установленный в собственном корпусе. При встряхивании или повороте корпуса, грузик перемещается внутри него по инерции. И поскольку при ускорении в соответствующем направлении грузик движется, он неизбежно натягивает за собой и пружинку, колебания которой могут быть приняты в расчет для определения направления и ускорения изменения положения всего корпуса. Так, три пружинки с грузиками, установленные вдоль трех пространственных осей, позволяют получить полное представление о направлении и величине ускорения относительно земли того предмета, к которому они прикреплены.

Гироскоп – это устройство, способное измерять угловую скорость вращения объекта вокруг его оси. Существуют различные типы гироскопов, такие как механические, оптические, микроэлектромеханические (MEMS) и др.

Назначение: Гироскопы используются в навигационных системах, авиации, мореплавании, робототехнике, стабилизации камер и других устройствах, где необходимо измерять и контролировать угловую скорость объекта.

Гироскоп действует совсем иначе. Волчок, закреплен на оси в первой рамке, и вращается с высокой скоростью. Первая рамка имеет свободу вращения, будучи закреплена на оси (перпендикулярной оси волчка) внутри второй рамки. Третья рамка несет на себе ось вращения для второй рамки (перпендикулярную осям вращения волчка и первой рамки).

В итоге, как ни поворачивай такую конструкцию, волчок будет стремиться сохранить вертикальное положение, хотя кольца и будут поворачиваться.

35. Ультразвуковой датчик расстояния. Принцип работы. Схема подключения.

Ультразвуковой датчик расстояния – это устройство, которое использует ультразвуковые волны для измерения расстояния до объекта. Датчик расстояния является прибором бесконтактного типа, и обеспечивает высокоточное измерение и стабильность. На его работу не оказывает существенного воздействия электромагнитные излучения и солнечная энергия.

Датчик оснащен четырьмя выводами (стандарт 2, 54 мм):

1. Контакт питания положительного типа – +5V;
2. Trig (T) – выход сигнала входа;
3. Echo (R) – вывод сигнала выхода;
4. GND – вывод «Земля».

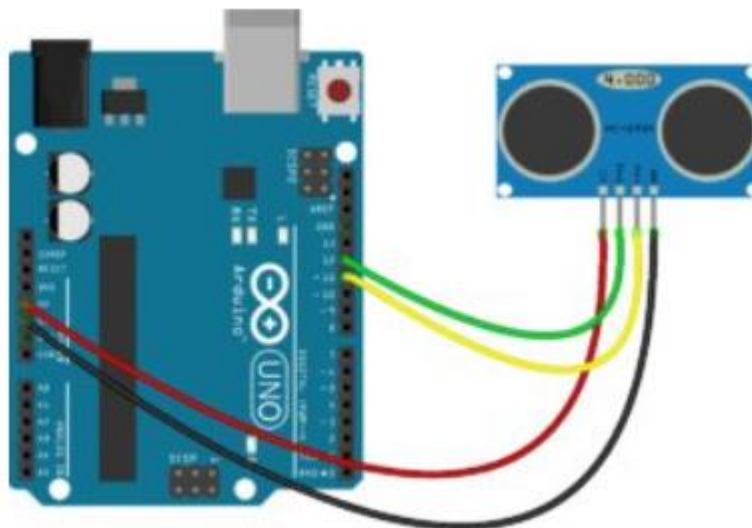
Принцип работы ультразвукового датчика расстояния основан на измерении времени, за которое ультразвуковой сигнал отправляется от датчика к объекту и обратно.

Принцип работы:

1. Ультразвуковой датчик генерирует ультразвуковые волны и отправляет их в сторону объекта.
2. Ультразвуковые волны отражаются от объекта и возвращаются обратно к датчику.
3. Датчик измеряет время, за которое ультразвуковой сигнал прошел от датчика до объекта и обратно.
4. Исходя из времени прохождения сигнала и скорости звука в воздухе, датчик рассчитывает расстояние до объекта.

Схема подключения и основные элементы:

1. Ультразвуковой датчик: подключается к микроконтроллеру или другому устройству управления.
2. Микроконтроллер: принимает данные от ультразвукового датчика и обрабатывает их для определения расстояния до объекта.
3. Источник питания: обеспечивает питание как для ультразвукового датчика, так и для микроконтроллера.



36. Инфракрасный датчик расстояния. Принцип работы. Схема подключения.

Инфракрасный датчик расстояния – это устройство, которое использует инфракрасные лучи для измерения расстояния до объекта.

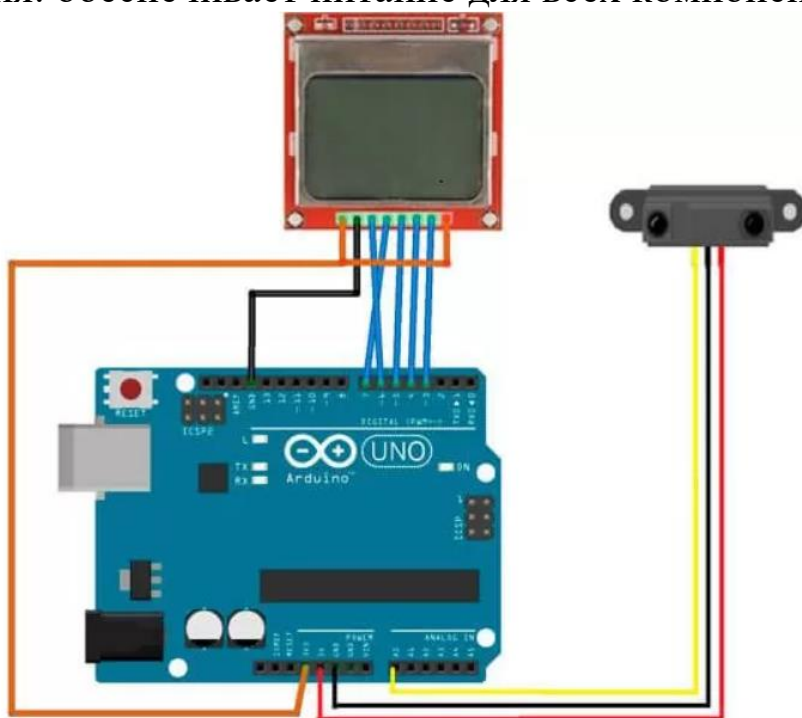
Принцип работы инфракрасного датчика расстояния основан на измерении времени задержки между отправкой инфракрасного сигнала и его отражением от объекта.

Принцип работы:

1. Инфракрасный датчик отправляет инфракрасный сигнал в сторону объекта.
2. Сигнал отражается от объекта и возвращается обратно к датчику.
3. Датчик измеряет время, за которое инфракрасный сигнал прошел от датчика до объекта и обратно.
4. Исходя из времени прохождения сигнала и скорости света, датчик рассчитывает расстояние до объекта.

Схема подключения и основные элементы:

5. Инфракрасный датчик: подключается к микроконтроллеру или другому устройству управления.
6. Источник инфракрасного излучения: генерирует инфракрасный сигнал, который отправляется к объекту.
7. Приемник инфракрасного излучения: принимает отраженный сигнал от объекта и передает его на обработку.
8. Микроконтроллер: обрабатывает данные от датчика и рассчитывает расстояние до объекта.
9. Источник питания: обеспечивает питание для всех компонентов системы.



37. Сферы применения беспроводных сетей. Преимущества беспроводных сетей.

Беспроводные сети – это сети связи, в которых передача данных осуществляется без использования проводов или кабелей. Вместо этого, данные передаются по радиоволнам, инфракрасному излучению или другим беспроводным технологиям.

Применение беспроводных сетей:

1. Домашнее использование - они позволяют пользователям свободно перемещаться по дому, не ограничиваясь проводами.
2. Бизнес и офисы - обеспечивают гибкость и мобильность сотрудников, позволяя им работать из любой точки офиса.
3. Общественные места - таких как кафе, рестораны, аэропорты, отели и торговые центры. Они позволяют посетителям подключаться к интернету и оставаться на связи во время пребывания в этих местах.
4. Транспорт - таком как поезда, автобусы и самолеты, чтобы предоставить пассажирам доступ к интернету во время путешествия.
5. Медицина - для передачи данных и обмена информацией между медицинскими устройствами, такими как мониторы пациентов, сканеры и медицинские аппараты.
6. Промышленность - используются для мониторинга и управления различными процессами и устройствами. Они позволяют собирать данные, контролировать оборудование и оптимизировать производственные процессы.

Преимущества беспроводных сетей:

1. Мобильность: позволяют пользователям подключаться к сети и обмениваться данными в любом месте, где есть доступ к сети.
2. Удобство установки: нет необходимости прокладывать провода и делать сложные настройки. Достаточно настроить беспроводной маршрутизатор и подключить устройства к сети.
3. Гибкость: беспроводные сети позволяют легко добавлять новые устройства в сеть или перемещать существующие устройства без необходимости переключать провода.
4. Снижение затрат: не требуют дополнительных затрат на проводное оборудование и его установку.

38. Классификация беспроводных технологий коммуникаций

По дальности действия:

1. Беспроводные персональные сети (WPAN — Wireless Personal Area Networks). Примеры технологий — Bluetooth.
2. Беспроводные локальные сети (WLAN — Wireless Local Area Networks). Примеры технологий — Wi-Fi.
3. Беспроводные сети масштаба города (WMAN — Wireless Metropolitan Area Networks). Примеры технологий — WiMAX.
4. Беспроводные глобальные сети (WWAN — Wireless Wide Area Network). Примеры технологий — CSD, GPRS, EDGE, EV-DO, HSPA.

По топологии:

1. «Точка-точка».
2. «Звезда».

По области применения:

1. Корпоративные (ведомственные) беспроводные сети — создаваемые компаниями для собственных нужд.
2. Операторские беспроводные сети — создаваемые операторами связи для возмездного оказания услуг.

Беспроводные технологии коммуникаций можно классифицировать по различным критериям, таким как диапазон частот, скорость передачи данных, дальность действия, способ модуляции сигнала и т. д.

1. По диапазону частот:

- Bluetooth: работает в диапазоне частот 2.4 ГГц.
- Wi-Fi: использует различные диапазоны частот, включая 2.4 ГГц и 5 ГГц.
- Zigbee: работает в диапазоне частот 2.4 ГГц.
- LTE (4G): использует различные диапазоны частот для передачи данных мобильной связи.

2. По дальности действия:

- Bluetooth: обычно имеет дальность действия до 10 метров.
- Wi-Fi: может обеспечивать дальность действия до нескольких сотен метров в зависимости от обстановки.
- LTE (4G): позволяет обеспечить связь на большие расстояния, до нескольких километров.

3. По скорости передачи данных:

- Bluetooth: обычно поддерживает скорость передачи данных до нескольких десятков мегабит в секунду.
- Wi-Fi: может обеспечивать высокие скорости передачи данных, до сотен мегабит в секунду.
- LTE (4G): позволяет достигать высоких скоростей передачи данных, до нескольких сотен мегабит в секунду.

4. По способу модуляции сигнала:

-ASK (Amplitude Shift Keying): изменение амплитуды сигнала для передачи информации.

-FSK (Frequency Shift Keying): изменение частоты сигнала для передачи информации.

-PSK (Phase Shift Keying): изменение фазы сигнала для передачи информации.

RFID — способ автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются данные, хранящиеся в так называемых транспондерах, или RFID-метках.

NFC (Near Field Communication) — технология беспроводной высокочастотной связи малого радиуса действия, которая дает возможность обмена данными между устройствами, находящимися на расстоянии около 10 сантиметров.

Bluetooth — производственная спецификация беспроводных gthcjyfkmys[cntntq Bluetooth обеспечивает обмен информацией между такими устройствами как персональные компьютеры (настольные, карманные, ноутбуки), мобильные телефоны, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики, наушники, гарнитуры на надёжной, бесплатной, повсеместно доступной радиочастоте для ближней связи.

UWB — это беспроводная технология связи на малых расстояниях при низких затратах энергии, использующая в качестве несущей сверхширокополосные сигналы с крайне низкой спектральной плотностью мощности.

DECT — технология беспроводной связи на частотах 1880—1900 МГц с модуляцией GMSK ($BT = 0,5$), используется в современных радиотелефонах.

GPRS — надстройка над технологией мобильной связи GSM, осуществляющая пакетную передачу данных. GPRS позволяет пользователю сети сотовой связи производить обмен данными с другими устройствами в сети GSM и с внешними сетями, в том числе Интернет.

IRDA (Infrared Data Association) — ИК-порт, Инфракрасный порт — группа стандартов, описывающая протоколы физического и логического уровня передачи данных с использованием инфракрасного диапазона световых волн в качестве среды передачи.

39. Диаграмма сетевых телекоммуникационных характеристик BWN.

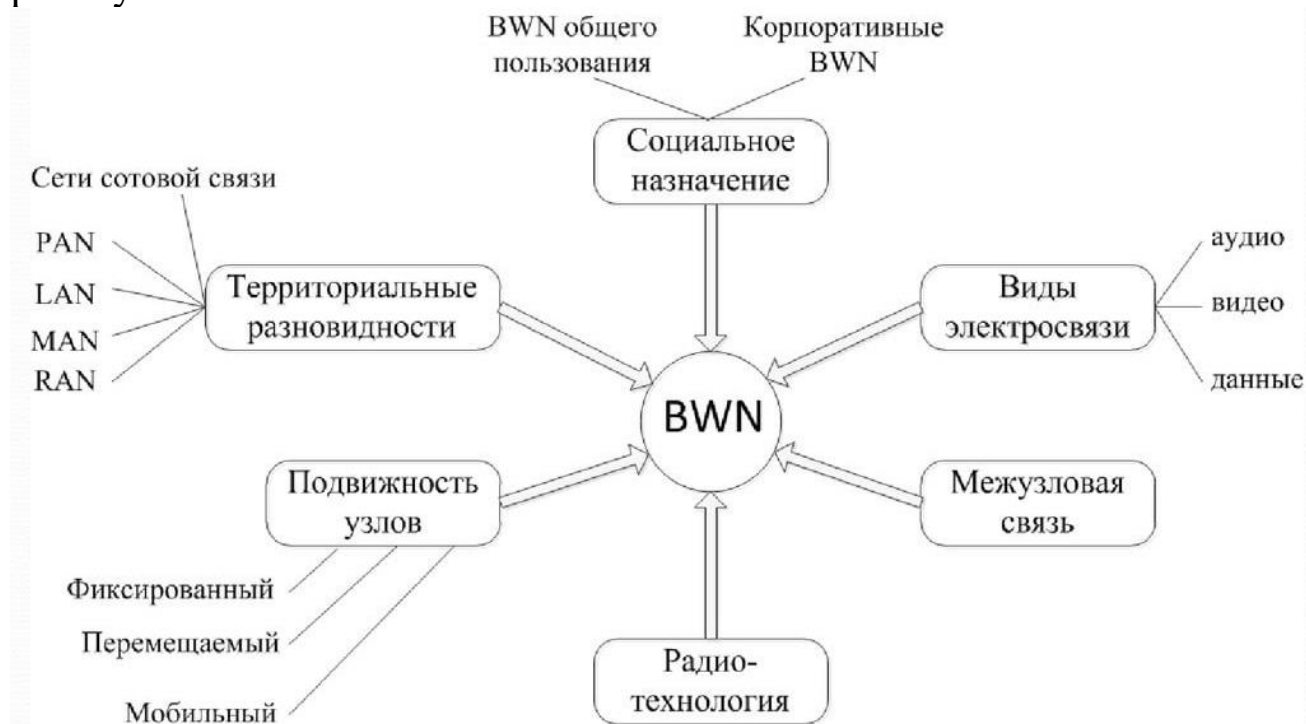
На данной диаграмме представлена структура и функции сети BWN (Broadband Wireless Network). Эта диаграмма помогает понять, как устроена и функционирует сеть BWN. В сети BWN (Broadband Wireless Network) маршрутизаторы, коммутаторы и серверы взаимодействуют, чтобы обеспечить эффективное функционирование сети.

Маршрутизаторы играют ключевую роль в сети BWN, они отвечают за маршрутизацию трафика между различными узлами сети. Маршрутизаторы принимают данные от источника, определяют наилучший путь для доставки данных и пересылают их по этому пути к назначению. Они также обеспечивают управление и контроль трафика в сети.

Коммутаторы в сети BWN выполняют функцию коммутации пакетов данных. Они подключают множество устройств внутри сети, таких как компьютеры, принтеры и другие устройства, и обеспечивают передачу данных между ними. Коммутаторы могут определить, на какой порт отправить данные на основе MAC-адреса устройства назначения. Они также обеспечивают более высокую скорость передачи данных по сравнению с хабами.

Серверы являются ключевыми компонентами сети BWN, предоставляющими различные сервисы пользователям. Они могут обеспечивать такие функции, как обмен файлами, электронная почта, доступ к веб-ресурсам и другие приложения. Серверы обрабатывают запросы от клиентов в сети и предоставляют им требуемую информацию или услуги.

Взаимодействие между маршрутизаторами, коммутаторами и серверами происходит путем обмена данными и управления трафиком. Маршрутизаторы определяют наилучший путь для доставки данных, коммутаторы обеспечивают передачу данных между устройствами, а серверы предоставляют необходимые услуги и обрабатывают запросы от клиентов. Эффективное взаимодействие между этими компонентами сети BWN обеспечивает стабильную и надежную работу сети.



40. Виды беспроводных сетей. Сети сотовой связи.

Виды беспроводных сетей:

Персональные беспроводные сети	<ul style="list-style-type: none">• IrDA — Infrared Data Association• Bluetooth; UWB — Ultra-Wide Band; Wireless USB• Wireless HD; WiGig; LibertyLink• WHDi — Wireless Home Digital Interface (Amimon)
Беспроводные сенсорные сети	<ul style="list-style-type: none">• DASH7; Z-Wave; Insteon; EnOcean• MiWi
Малые локальные беспроводные сети	<ul style="list-style-type: none">• HiperLAN — High Performance Radio LAN• Wi-Fi; Zigbee
Крупные локальные беспроводные сети	<ul style="list-style-type: none">• WiMAX — Worldwide Interoperability for Microwave Access• HiperMAN; WiBro — Wireless Broadband
Глобальные беспроводные сети	<ul style="list-style-type: none">• NMT — Nordic Mobile Telephone• AMPS — Advanced Mobile Phone System• Mobitex; CDMA — Code Division Multiple Access• LTE — Long-Term Evolution

Сети сотовой связи:

1. GSM (Global System for Mobile Communications): является одной из первых технологий сотовой связи и предоставляет услуги 2G. Он используется для передачи голосовых вызовов и SMS. GSM работает в различных диапазонах частот и обеспечивает международную роуминговую связь.

2. UMTS (Universal Mobile Telecommunications System): является технологией 3G и предоставляет более высокие скорости передачи данных по сравнению с 2G. Он поддерживает интернет, видеозвонки и другие мультимедийные услуги. UMTS использует сеть WCDMA (Wideband Code Division Multiple Access) для передачи данных.

3. LTE (Long-Term Evolution): является технологией 4G и обеспечивает еще более высокие скорости передачи данных по сравнению с 3G. Он поддерживает широкополосный доступ в интернет, потоковое видео и другие требовательные приложения. LTE использует IP-сеть для передачи данных.

4. 5G (Fifth Generation): является самой новой технологией сотовой связи, которая обеспечивает еще более высокие скорости передачи данных, низкую задержку и большую пропускную способность. Он позволяет подключать большое количество устройств и поддерживает новые технологии, такие как виртуальная и дополненная реальность.

41. Беспроводные платформы. Методы доступа.

существует несколько ключевых платформ, каждая из которых имеет свои особенности и методы доступа:

1. Wi-Fi (Wireless Fidelity):

- **Технология:** Использует радиоволны для передачи данных между устройствами.
- **Методы доступа:** Применяет стандарт IEEE 802.11, работает на частотах 2.4 ГГц и 5 ГГц.
- **Применение:** Широко используется для подключения к интернету в домах, офисах и общественных местах.

2. Bluetooth:

- **Технология:** Предназначена для беспроводной передачи данных на короткие расстояния.
- **Методы доступа:** Работает на частоте 2.4 ГГц, соединяет устройства напрямую для обмена данными.
- **Применение:** Используется для подключения гарнитур, мышей, клавиатур и других периферийных устройств.

3. Мобильная связь:

- **Технология:** Обеспечивает доступ в интернет с помощью сотовых телефонов и других мобильных устройств.
- **Методы доступа:** Включает в себя различные поколения сетей, такие как 3G, 4G/LTE и 5G, которые используют разные частоты и технологии для передачи данных.

Применение: Основа для мобильного интернета и телефонной связи.

Каждая из этих платформ использует свои методы множественного доступа, такие как TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access), CDMA (Code Division Multiple Access), и другие, чтобы обеспечить эффективное использование доступного спектра и минимизировать помехи между устройствами.

42. Эволюция систем мобильной сотовой связи. 1946 – 1991.

1946 – Появление мобильной связи:

Первые принципы мобильной связи были заложены компаниями AT&T и Bell Telephone Laboratories.

Устройства весили более 30 кг и имели отдельный блок питания.

1950-е – Радиотелефоны:

В это время использовались радиотелефоны, которые позволяли водителям автомобилей звонить с помощью оператора.

1960-е – Улучшение технологий:

Были разработаны более компактные и удобные устройства.

Появление автоматических систем, которые не требовали вмешательства оператора для установления соединения.

1970-е – Первые сотовые сети:

В 1973 году Motorola представила первый портативный сотовый телефон.

В 1979 году в Японии была запущена первая коммерческая сотовая сеть.

1980-е – Первое поколение (1G):

В 1983 году в США была запущена первая сотовая сеть 1G, которая использовала аналоговые технологии.

Сети 1G позволяли только голосовую связь и имели ограниченное покрытие.

1990-е – Переход к цифровым технологиям:

В начале 1990-х годов начался переход от аналоговых к цифровым технологиям.

В 1991 году была опубликована спецификация GSM, которая стала основой для современных мобильных сетей.

43. Эволюция систем мобильной сотовой связи. 1991 – 2007+.

1991 – Рождение 2G и GSM:

В этом году был представлен стандарт **GSM** (Global System for Mobile Communications), который впервые использовал цифровую передачу данных и сигнализацию.

GSM ввел такие понятия, как SIM-карты, шифрование данных и международный роуминг.

1992 – SMS и первые сообщения:

Стандарт GSM позволил отправлять короткие текстовые сообщения (SMS), что стало революцией в общении.

1998 – Появление GPRS:

GPRS (General Packet Radio Service) был введен как часть стандарта GSM, позволяя передачу данных в режиме пакетной коммутации.

Это улучшило скорость передачи данных и позволило пользователям оставаться онлайн постоянно.

2001 – Запуск 3G сетей:

Стандарт **UMTS** (Universal Mobile Telecommunications System) был запущен, предложив более высокие скорости передачи данных и улучшенные услуги, такие как видеозвонки.

3G сети использовали технологию **WCDMA** (Wideband Code Division Multiple Access), которая обеспечивала большую пропускную способность.

2003 – EDGE:

EDGE (Enhanced Data rates for GSM Evolution) был представлен как улучшение для сетей 2G, предлагая более высокие скорости передачи данных.

2007 и далее – Начало эры смартфонов:

С появлением iPhone в 2007 году началась новая эра смартфонов, которые требовали ещё более высоких скоростей и стабильности связи.

Это стимулировало разработку и внедрение технологий 4G, таких как **LTE** (Long-Term Evolution), которые начали внедряться в конце 2000-х годов. Этот период был временем быстрого технологического прогресса, который привел к значительному улучшению мобильной связи и положил начало современной эре мобильного интернета.

44. Эволюция систем высокоскоростного доступа. Перечень рабочих групп IEEE 802.

Эволюция систем высокоскоростного доступа и рабочие группы IEEE 802 играют ключевую роль в развитии сетевых технологий.

Эволюция систем высокоскоростного доступа: Системы высокоскоростного доступа развивались от простых модемных соединений до сложных многоуровневых сетей. Основные этапы включают:

WiMAX: Технология для широкополосного беспроводного доступа, предшественник 4G.

Wi-Fi: Стандарт IEEE 802.11, обеспечивающий беспроводной доступ в локальных сетях.

UMTS и EV-DO: Стандарты для мобильных сетей, предшественники LTE и 5G.

Перечень рабочих групп IEEE 802: IEEE 802 — это семейство стандартов, касающихся локальных и городских сетей. Вот некоторые из рабочих групп:

IEEE 802.1: Управление сетевыми устройствами.

IEEE 802.3: Ethernet.

IEEE 802.11: Беспроводные локальные сети (WLAN), Wi-Fi.

IEEE 802.15: Беспроводные персональные сети (WPAN), включая Bluetooth.

IEEE 802.16: Беспроводные городские сети (WiMAX).

Эти рабочие группы разрабатывают и обновляют стандарты, которые определяют функционирование сетевых технологий на физическом и канальном уровнях, обеспечивая совместимость и эффективность сетевых операций.

Эти стандарты и рабочие группы способствуют постоянному улучшению и развитию сетевых технологий, что позволяет достигать более высоких скоростей передачи данных и обеспечивать более надежные и безопасные сетевые соединения.

45. ОС реального времени. Признаки систем жесткого и мягкого реального времени.

Операционные системы реального времени (ОСРВ) играют важную роль в управлении процессами и задачами, которые требуют строгого соблюдения временных ограничений.

ОС реального времени (ОСРВ): ОСРВ — это специализированные операционные системы, предназначенные для обработки данных в режиме реального времени, часто без задержек. Они обеспечивают гарантированное время отклика на запросы и управляют внешними событиями, процессами или объектами

Системы жесткого реального времени:

- **Определение:** Системы, которые должны гарантировать выполнение задач в строго определенных временные рамки, даже в худших условиях.
- **Признаки:**
 - Гарантированное время реакции на внешние события.
 - Жесткая подсистема планирования процессов, где высокоприоритетные задачи не вытесняются низкоприоритетными.
 - Недопустимость задержек, так как это может привести к серьезным последствиям, включая аварии и катастрофы.
 - Ситуация, когда обработка событий происходит за время, большее предусмотренного, считается фатальной ошибкой

Системы мягкого реального времени:

- **Определение:** Системы, которые стремятся обеспечить выполнение задач в реальном времени в среднем, но не всегда гарантируют это.
- **Признаки:**
 - Задержка реакции считается восстановимой ошибкой, которая может привести к увеличению стоимости результатов и снижению производительности, но не является фатальной.
 - Примером может служить работа компьютерной сети, где необработанные пакеты могут быть повторно отправлены, что снижает производительность, но не приводит к потере данных

ОСРВ обычно используются в критически важных системах, где требуется надежность и точность, например, в медицинских устройствах, автомобильных системах, системах управления полетами и промышленном контроле.

46. Определение операционной системы реального времени. Требования. Критические ресурсы.

Операционная система реального времени (ОСРВ) — это специализированный тип операционной системы, предназначенный для управления аппаратными и программными ресурсами в такой манере, чтобы гарантировать выполнение критических задач в определённые временные рамки. ОСРВ обеспечивает предсказуемое и надёжное время отклика, что критически важно для систем, где задержка или пропуск срока выполнения может привести к неприемлемым последствиям.

Требования к ОСРВ:

- **Многозадачность:** Поддержка одновременного выполнения множества задач.
- **Вытеснение:** Возможность приостановки текущей задачи для выполнения более приоритетной.
- **Приоритеты:** Наличие системы приоритетов для определения порядка выполнения задач.
- **Синхронизация:** Предсказуемые механизмы синхронизации для предотвращения конфликтов доступа к ресурсам.
- **Наследование приоритетов:** Механизмы для предотвращения проблемы приоритетного инверсии.

Критические ресурсы в ОСРВ: Критические ресурсы — это те компоненты системы, которые имеют ограниченное предложение и высокий спрос, особенно в условиях жёстких временных ограничений. В ОСРВ к таким ресурсам относятся:

- **Процессорное время:** Ограниченное количество времени процессора, доступное для выполнения всех задач.
- **Память:** Ограниченный объём оперативной и постоянной памяти для хранения и выполнения программ.
- **Устройства ввода/вывода:** Ограниченное количество устройств ввода/вывода и каналов связи, которые должны быть эффективно распределены между задачами.
- **Данные:** Информация, необходимая для выполнения задач, которая может быть ограничена или должна быть защищена от несанкционированного доступа.

Эти ресурсы требуют тщательного планирования и управления в ОСРВ, чтобы обеспечить своевременное выполнение всех критических задач и поддержание стабильности системы.

47. Требования, накладываемые на вычислительную установку реального времени.

Вычислительная установка реального времени должна соответствовать ряду строгих требований, чтобы обеспечить надежное и своевременное выполнение задач. Основные требования:

1. **Детерминированное время отклика:** Система должна гарантировать выполнение задач в предсказуемые и строго определенные временные рамки.
2. **Предсказуемость:** Поведение системы, включая время обработки и время отклика, должно быть предсказуемым.
3. **Надежность:** Система должна быть устойчива к сбоям и способна восстанавливаться после ошибок.
4. **Синхронизация:** Система должна поддерживать точную синхронизацию между различными задачами и процессами.
5. **Многозадачность:** Поддержка одновременного выполнения нескольких задач с различными приоритетами.
6. **Управление ресурсами:** Эффективное распределение и управление ограниченными ресурсами, такими как процессорное время, память и устройства ввода/вывода.
7. **Безопасность:** Защита от несанкционированного доступа и обеспечение конфиденциальности данных.
8. **Тестирование и отладка:** Возможность проведения тщательного тестирования и отладки для обеспечения корректной работы системы.
9. **Соответствие стандартам:** Соблюдение международных и отраслевых стандартов, касающихся систем реального времени.

Эти требования обеспечивают, то что вычислительная установка реального времени может надежно функционировать в критически важных приложениях, где задержки или сбои могут привести к серьезным последствиям.

48. Области применения систем реального времени.

Системы реального времени (СРВ) находят применение в самых разнообразных областях, где требуется немедленная обработка данных и быстрый отклик. Области применения СРВ:

1. **Промышленное управление:** Автоматизированные системы управления технологическими процессами, такие как химические или ядерные реакторы, требуют СРВ для мониторинга и контроля процессов в реальном времени.
2. **Транспорт:** Бортовые компьютеры в автомобилях, самолетах, кораблях и других транспортных средствах используют СРВ для управления движением, навигации и безопасности.
3. **Медицинские устройства:** СРВ применяются в критически важных медицинских устройствах, таких как аппараты искусственной вентиляции легких, мониторы сердечного ритма и системы хирургической навигации.
4. **Телекоммуникации:** СРВ необходимы для управления сетевым оборудованием, обеспечивая надежную и своевременную передачу данных.
5. **Оборонная промышленность:** Военные системы, включая радары и системы управления оружием, полагаются на СРВ для быстрого реагирования на угрозы и управления сложными операциями.
6. **Аэрокосмическая отрасль:** СРВ используются в бортовых системах управления пилотируемыми и беспилотными космическими аппаратами для выполнения задач управления и сбора данных в космосе.
7. **Развлекательная индустрия:** Системы обработки аудио- и видеопотоков при трансляции в прямом эфире также используют СРВ для обеспечения синхронизации и минимальной задержки.

СРВ обеспечивают надежность и эффективность в ситуациях, где время является критическим фактором.

\

49. Вычислительные установки, на которых применяются СРВ.

Вычислительные установки, на которых применяются системы реального времени (СРВ), охватывают широкий спектр устройств и платформ, каждая из которых предназначена для выполнения определенных задач в строгом соответствии с временными ограничениями.

1. **Промышленные контроллеры:** Используются для автоматизации и контроля производственных процессов, таких как химические или ядерные реакторы, а также управления теплоэлектростанциями.
2. **Бортовые системы управления:** Применяются в пилотируемых и беспилотных летательных и космических аппаратах для управления полетом и выполнения миссий.
3. **Транспортные средства:** Автомобили, корабли и другие виды транспорта используют СРВ для управления двигателем, навигацией и системами безопасности.
4. **Военное и оборонное оборудование:** Радары, системы управления оружием, ракеты и другие военные технологии зависят от СРВ для своевременного и точного реагирования на изменения ситуации.
5. **Серверные фермы и сети хранения данных:** Используются для обработки больших объемов данных с высокой скоростью и надежностью.
6. **Бытовая техника:** Некоторые виды бытовой техники, такие как стиральные машины и микроволновые печи, также могут использовать СРВ для управления своими функциями.
7. **Обычные компьютеры:** Некоторые настольные компьютеры могут быть настроены для работы с ОСРВ для специализированных задач, таких как аудио- или видеообработка в реальном времени.

Эти примеры показывают, что СРВ могут быть интегрированы в самые разные устройства и платформы, обеспечивая надежность и точность в критически важных приложениях.

50. ОСРВ с монолитной архитектурой.

Операционные системы реального времени (ОСРВ) с монолитной архитектурой представляют собой классический подход к разработке операционных систем, где все компоненты системы тесно интегрированы в единый блок. ОСРВ с монолитной архитектурой:

Определение монолитной архитектуры: Монолитная архитектура ОСРВ означает, что все основные сервисы и функции системы, такие как управление памятью, планирование задач, обработка прерываний и драйверы устройств, собраны вместе в одном большом блоке кода или ядре. Это противоположность модульному или микросервисному подходу, где различные функции разделены на отдельные компоненты или сервисы.

Преимущества:

- **Простота разработки и отладки:** Так как все компоненты тесно связаны, разработчикам проще понять взаимодействие между различными частями системы.
- **Высокая производительность:** Монолитные ОСРВ могут обеспечивать более высокую производительность, поскольку они избегают накладных расходов на взаимодействие между модулями.

Недостатки:

- **Сложность масштабирования:** По мере роста системы, монолитная архитектура может стать трудной для управления и масштабирования.
- **Трудности в обновлении:** Обновление одного компонента часто требует перекомпиляции и развертывания всей системы.
- **Уязвимость:** Ошибка в одном компоненте может повлиять на всю систему, что потенциально может привести к сбоям.

Примеры ОСРВ с монолитной архитектурой:

- **VxWorks:** Широко используемая в аэрокосмической и оборонной промышленности.
- **QNX:** Применяется в автомобильной промышленности и для критически важных систем.

Монолитные ОСРВ по-прежнему активно используются во многих областях, где требуется высокая надежность и производительность, несмотря на некоторые трудности в управлении и масштабировании.

51. ОСРВ на основе ядра.

Операционные системы реального времени (ОСРВ) на основе ядра представляют собой архитектуру, где ядро системы выполняет основные функции, такие как управление памятью, планирование задач и обработка прерываний.

Ядро ОСРВ: Ядро ОСРВ — это центральный компонент системы, который обеспечивает низкоуровневое взаимодействие с аппаратным обеспечением и предоставляет основные сервисы для выполнения прикладных программ. Ядро отвечает за следующие функции:

- **Управление памятью:** Выделение и освобождение памяти для процессов и задач.
- **Планирование задач:** Определение порядка выполнения задач в соответствии с их приоритетами.
- **Синхронизация и взаимодействие задач:** Предоставление механизмов для синхронизации и обмена данными между задачами.
- **Обработка прерываний:** Реагирование на события аппаратного обеспечения и прерывания для обеспечения своевременного выполнения задач.

Типы ядер ОСРВ:

- **Монолитное ядро:** Все сервисы и драйверы включены непосредственно в ядро.
- **Микроядро:** Основные сервисы находятся в ядре, а дополнительные функции выполняются в пользовательском пространстве.
- **Гибридное ядро:** Сочетает элементы монолитного и микроядра, обеспечивая баланс между производительностью и модульностью.

Примеры ОСРВ на основе ядра:

- **FreeRTOS:** Широко используемая ОСРВ с микроядром, предназначенная для встраиваемых систем
- **VxWorks:** Монолитная ОСРВ, применяемая в критически важных системах, таких как аэрокосмическая и оборонная промышленность.
- **QNX:** Микроядро, используемое в автомобильной промышленности и других областях, где требуется высокая надежность.

ОСРВ на основе ядра выбираются для разработки систем, где требуется строгое соблюдение временных ограничений и высокая надежность.

52. Объектно-ориентированная ОСРВ.

Объектно-ориентированные операционные системы реального времени (ОСРВ) используют принципы объектно-ориентированного программирования (ООП) для улучшения модульности, масштабируемости и повторного использования кода в системах реального времени.

Основные принципы ООП, применяемые в ОСРВ:

- **Инкапсуляция:** Скрытие внутреннего состояния объекта от внешнего мира и предоставление интерфейса для взаимодействия с объектом.
- **Наследование:** Создание новых классов на основе существующих, что позволяет переиспользовать и расширять функциональность.
- **Полиморфизм:** Возможность обращения с объектами разных классов через интерфейс их базового класса.

Преимущества объектно-ориентированных ОСРВ:

- **Повышение надежности:** Благодаря инкапсуляции и строгому определению интерфейсов, вероятность ошибок снижается.
- **Упрощение разработки:** Наследование и полиморфизм упрощают добавление новых функций и устройств.
- **Легкость обслуживания:** Модульность и возможность повторного использования кода облегчают поддержку и обновление системы.

Примеры объектно-ориентированных ОСРВ:

- **QNX Neutrino:** Микроядерная ОСРВ, использующая объектно-ориентированный подход для обеспечения модульности и безопасности.
- **OSEK/VDX:** Стандарт для встраиваемых операционных систем в автомобильной промышленности, поддерживающий объектно-ориентированные концепции.

Объектно-ориентированные ОСРВ обеспечивают гибкость и расширяемость, что делает их идеальными для сложных систем реального времени, где требуется быстрая адаптация к изменяющимся техническим требованиям и условиям эксплуатации.

53. ОС реального времени FreeRTOS. Основные характеристики. Применения.

FreeRTOS — это популярная операционная система реального времени, предназначенная для встраиваемых систем.

Основные характеристики FreeRTOS:

- **Малый размер:** FreeRTOS разработана для работы на микроконтроллерах с ограниченными ресурсами, такими как низкое быстродействие и малый объем оперативной и постоянной памяти
- **Простота:** Ядро системы просто в использовании и может быть легко адаптировано под конкретные нужды проекта.
- **Многозадачность:** Поддерживает простую многозадачность с приоритетами задач, что позволяет эффективно управлять выполнением нескольких процессов.
- **Модульность:** Предоставляет широкий спектр модулей и библиотек для расширения функциональности.
- **Портативность:** Поддерживает множество архитектур микроконтроллеров и легко переносится между различными платформами.

Применения FreeRTOS:

- **Встраиваемые системы:** Используется в различных встраиваемых приложениях, от простых бытовых устройств до сложных промышленных систем управления.
- **Интернет вещей (IoT):** Подходит для устройств IoT благодаря своей способности работать на устройствах с ограниченными ресурсами и низким энергопотреблением.
- **Робототехника:** Применяется в роботах для управления движением, обработки сенсорных данных и выполнения задач в реальном времени.
- **Автомобильная промышленность:** Используется в автомобильных системах для управления двигателем, системами безопасности и навигации.

FreeRTOS предлагает гибкость и надежность для разработки систем реального времени, требующих строгого соблюдения временных ограничений и высокой стабильности работы.

54. ОС реального времени QNX. Основные характеристики. Применения.

Операционная система реального времени QNX — это мощная и надежная система, разработанная для широкого спектра устройств, от встраиваемых систем до крупномасштабных распределенных сетей.

Основные характеристики QNX:

- **Микроядро:** QNX использует микроядерную архитектуру, что обеспечивает высокую модульность и надежность.
- **POSIX-совместимость:** Это обеспечивает совместимость с множеством стандартов UNIX, что упрощает портирование приложений.
- **Масштабируемость:** QNX подходит как для простых однопроцессорных систем, так и для сложных многопроцессорных конфигураций.
- **Безопасность:** ОС предлагает различные функции для обеспечения безопасности и изоляции процессов.
- **Возможность реального времени:** QNX обеспечивает строгие гарантии времени отклика, что критически важно для систем реального времени.

Применения QNX:

- **Автомобильная промышленность:** QNX широко используется в информационно-развлекательных системах автомобилей и системах помощи водителю.
- **Промышленное управление:** ОС применяется для управления промышленными процессами, включая робототехнику и автоматизацию.
- **Медицинские устройства:** QNX находит применение в критически важных медицинских системах, таких как аппараты искусственной вентиляции легких.
- **Телекоммуникации:** ОС используется в оборудовании для связи, включая маршрутизаторы и коммутаторы.
- **Аэрокосмическая отрасль:** QNX применяется в бортовых системах управления летательных аппаратов и спутников.

QNX является одной из наиболее надежных и проверенных операционных систем для встраиваемых систем, предлагая разработчикам гибкость и мощные инструменты для создания систем с критически важными требованиями к времени отклика.

55.	Мультипрограммирование.	Характерные	показатели
эффективности вычислительных систем.			

Мультипрограммирование, также известное как многозадачность, — это метод организации вычислительного процесса, при котором несколько программ могут выполняться на одном процессоре поочередно, используя общие ресурсы системы, такие как процессор, оперативная и внешняя память, устройства ввода-вывода и данные. Этот подход направлен на повышение эффективности использования вычислительных систем.

Характерные показатели эффективности вычислительных систем:

- **Пропускная способность:** Количество задач, выполняемых системой за единицу времени.
- **Удобство работы пользователей:** Возможность интерактивной работы с несколькими приложениями одновременно.
- **Реактивность системы:** Способность системы выдерживать заранее заданные интервалы времени между запуском программы и получением результата.

В зависимости от выбранного критерия эффективности, операционные системы могут быть разделены на системы пакетной обработки, системы разделения времени и системы реального времени. Каждый тип имеет свои особенности и области применения.

Мультипрограммирование позволяет более эффективно использовать ресурсы компьютера, уменьшая время простоя и увеличивая общую производительность системы.

56. Подсистема управления процессами и потоками. Создание процесса.

Подсистема управления процессами и потоками является ключевой частью операционных систем, особенно в многозадачных средах. Она отвечает за создание, планирование и управление процессами и потоками, что позволяет эффективно использовать ресурсы вычислительной системы.

Создание процесса включает в себя следующие этапы:

1. **Присвоение идентификатора:** Каждому новому процессу присваивается уникальный идентификатор (PID).
2. **Выделение памяти:** Для процесса выделяется необходимое пространство в памяти, включая место для кода программы, данных и стека.
3. **Инициализация блока управления процессом (PCB):** Создается и инициализируется структура данных, содержащая всю информацию о процессе, необходимую для его управления.
4. **Добавление в очередь готовых к выполнению:** Процесс помещается в очередь процессов, ожидающих назначения на процессор.

Подсистема управления процессами также отвечает за **планирование**, то есть определение порядка, в котором процессы будут получать доступ к процессору, и за **синхронизацию**, обеспечивая корректное взаимодействие между процессами и потоками.

Эффективность подсистемы управления процессами напрямую влияет на общую производительность системы, поскольку она определяет, как быстро и эффективно система может реагировать на задачи пользователя и системные события.

57. Переход от выполнения одного потока к другому. Завершение процесса.

Переход от выполнения одного потока к другому, также известный как **переключение контекста**, является ключевым механизмом в многозадачных операционных системах. Этот процесс позволяет процессору переключаться между различными потоками (или процессами), чтобы обеспечить их кажущееся одновременное выполнение.

Процесс переключения контекста включает:

1. Сохранение контекста текущего потока, включая состояние регистров процессора и другую критически важную информацию.
2. Выбор следующего потока для выполнения, что обычно определяется планировщиком ОС на основе приоритетов и политик планирования.
3. Восстановление контекста нового потока, чтобы процессор мог продолжить его выполнение с того места, где он был остановлен.

Завершение процесса включает освобождение всех системных ресурсов, которые были выделены процессу, включая память и дескрипторы файлов. Процесс может быть завершен нормально, когда он выполнил свои задачи, или принудительно, если возникает необходимость в его остановке из-за ошибок или заморозки. В операционных системах, таких как Windows, для завершения процессов можно использовать команду `taskkill` в командной строке.

58. Основные данные в контексте процесса. Состояние процесса. Очереди процессов.

Контекст процесса, состояние процесса и очереди процессов — это ключевые понятия в управлении процессами операционной системы.

Основные данные в контексте процесса: Контекст процесса включает всю информацию, необходимую операционной системе для управления процессом. Это включает:

- **Пользовательский контекст:** Содержит код программы, данные и стек.
- **Регистровый контекст:** Включает состояние всех регистров процессора для данного процесса.
- **Системный контекст:** Содержит информацию о системных ресурсах, которые использует процесс.

Состояние процесса: Процесс в операционной системе может находиться в одном из нескольких состояний:

- **Новый:** Процесс создан, но еще не готов к выполнению.
- **Готовый:** Процесс загружен в память и ожидает назначения процессора.
- **Выполняется:** Процесс активно использует процессор.
- **Ожидание:** Процесс не может продолжить выполнение до наступления определенного события.
- **Завершенный:** Процесс выполнил свои задачи или был прерван.

Очереди процессов: Очереди процессов используются для организации процессов в зависимости от их состояния. Существуют различные типы очередей:

- **Очередь заданий:** Хранит все процессы в системе.
- **Готовая очередь:** Содержит процессы, находящиеся в памяти и готовые к выполнению.
- **Очереди устройств:** Для процессов, ожидающих операций ввода-вывода.

59. Реализация пакета потоков (ULT). Преимущества. Недостатки.

Реализация пакета потоков на уровне пользователя (User-Level Threads, ULT) представляет собой один из подходов к многопоточности в операционных системах. В этой модели управление потоками осуществляется без прямого участия ядра ОС, что имеет свои преимущества и недостатки.

Преимущества ULT:

- **Быстрое переключение контекста:** Поскольку все операции с потоками происходят в пространстве пользователя, переключение контекста между потоками выполняется быстрее, чем в модели с потоками на уровне ядра.
- **Эффективное распределение ресурсов:** Пользователь может оптимизировать планирование и управление потоками в соответствии с потребностями приложения.
- **Независимость от ОС:** Потоки ULT не зависят от реализации многопоточности в конкретной операционной системе, что облегчает переносимость приложений.

Недостатки ULT:

- **Отсутствие поддержки ядра:** Если один поток блокируется, например, из-за операции ввода-вывода, вся программа может остановиться, так как ядро ОС не знает о существовании других потоков внутри процесса.
- **Проблемы с распределением процессорного времени:** Поскольку ядро видит только один процесс, оно может неравномерно распределять процессорное время между потоками, что приводит к неэффективности.
- **Сложности с синхронизацией:** Синхронизация и взаимодействие между потоками ULT могут быть более сложными для реализации, поскольку стандартные механизмы синхронизации ядра не применимы.

ULT подходит для приложений, где необходимо быстрое переключение контекста и нет требований к интенсивному взаимодействию с операционной системой. Однако, в случаях, когда требуется надежная многопоточность с полной поддержкой ядра, предпочтительнее использовать потоки на уровне ядра (Kernel-Level Threads, KLT).

60. Реализация пакета потоков (KLT). Преимущества. Недостатки.

Реализация пакета потоков на уровне ядра (Kernel-Level Threads, KLT) — это метод управления потоками, при котором все операции по созданию, планированию и управлению потоками выполняются непосредственно ядром операционной системы.

Преимущества KLT:

- **Поддержка ядра:** Потоки полностью управляются ядром, что позволяет им в полной мере использовать возможности многопроцессорности и многозадачности.
- **Ответственность за блокировку:** Если один поток блокируется, например, при операции ввода-вывода, другие потоки того же процесса могут продолжать выполняться.
- **Эффективное распределение времени:** Ядро может более эффективно распределять процессорное время между потоками, учитывая их приоритеты и требования.

Недостатки KLT:

- **Затраты на ресурсы:** Управление потоками на уровне ядра требует больше системных ресурсов, таких как память и процессорное время.
- **Медленное переключение контекста:** Переключение контекста между потоками KLT занимает больше времени, чем между потоками ULT, поскольку требует вмешательства ядра.
- **Сложность:** Реализация и управление потоками KLT сложнее, чем ULT, из-за необходимости взаимодействия с ядром для всех операций.

KLT обеспечивает более тесную интеграцию с операционной системой и лучше подходит для приложений, требующих высокой производительности и надежности многопоточности. Однако это также связано с увеличением накладных расходов и сложности управления потоками.